Eletrônica Digital

Lógica Combinacional e Seqüencial

Conselho Regional do SENAI-CE

Jorge Parente Frota Júnior
Presidente

Ivan Rodrigues Bezerra
Vice-Presidente

Alexandre Pereira Silva
João Fernandes Fontenelle
Francisco de Assis Alves de Almeida
Delegados das Atividades Industriais

Hermano Frank Júnior
José Fernando Castelo Banco Ponte
Marcos Pinheiro de Oliveira Cavalcante
Suplentes dos Delegados das Atividades Industriais

Samuel Brasileiro Filho
Representante do Ministério da Educação e Cultura

Franco de Magalhães Neto
Suplente do Ministério da Educação e Cultura

Alberto Fernandes de Farias Neto
Representante do Ministério do Trabalho

José Nunes Passos
Suplente do Ministério do Trabalho

Departamento Regional do SENAI-CE

Francisco das Chagas Magalhães
Diretor Regional

Cid Fraga
Gerente do Centro de Formação Profissional Waldyr Diogo de Siqueira

FIEC
SENAI

Federação das Indústrias do Estado do Ceará
Serviço Nacional de Aprendizagem Industrial
Departamento Regional do Ceará
Centro de formação Profissional
Waldyr Diogo de Siqueira

Eletrônica Digital

Lógica Combinacional e Seqüencial

Fortaleza-Ceará
2004

© 2004. SENAI. Departamento Regional do Ceará
Qualquer parte desta obra poderá ser reproduzida, desde que citada a fonte.

SENAI/CE
Centro de Formação Profissional Waldyr Diogo de Siqueira – CFP WDS
Núcleo de Educação Profissional – NEP

Este projeto foi elaborado por colaboradores desta unidade de negócios cujos nomes estão relacionados na folha de créditos.

Ficha Catalográfica

S474 SENAI. CE. CFP. WDS. **Eletrônica Digital.** Fortaleza, 2004. 177p. il

1 ELETRÔNICA DIGITAL I TÍTULO

CDU 621.3

SENAI
Serviço Nacional
de Aprendizagem
Industrial

Departamento
Regional do Ceará

Av. Francisco Sá, 7221
Barra do Ceará
60.310-003 – Fortaleza – Ceará
Telefax: (85) 485-7888
e-mail: senai-wds@sfiec.org.br

SUMÁRIO

1. SISTEMA DE NUMERAÇÃO 10
1.1 Introdução 10
1.2 O Sistema Binário de Numeração 10
1.3 Odômetro Decimal 10
1.4 Odômetro Binário 11
1.5 Conversão do Sistema Binário para o Sistema Decimal 13
1.6 Conversão do Sistema Decimal para o Sistema Binário 14
1.7 O Sistema Octal de Numeração 16
1.8 Conversão do Sistema Octal para o Sistema Decimal 17
1.9 Conversão do Sistema Decimal para o Sistema Octal 18
1.10 O Sistema Hexadecimal de Numeração 18
1.11 Conversão do Sistema Hexadecimal para o Sistema Decimal 19
1.12 Conversão do Sistema Decimal para o Sistema Hexadecimal 20

2. FUNÇÕES E PORTAS LÓGICAS 21
2.1 Introdução 21
2.2 Funções Lógicas E, OU, NÃO NE e NOU 21
2.3 Função E OU AND 22
2.4 Função OU ou OR 24
2.5 Função NÃO ou NOT 25
2.6 Função NÃO E, NOU ou NAND 26
2.7 Função NÃO OU, NOU ou NOR 27
2.8 Bloco OU EXCLUSIVO 28
2.9 Bloco COINCIDÊNCIA 29
2.10 Quadro Resumo 30

3. ÁLGEBRA DE BOOLE E SIMPLIFICAÇÃO DE CIRCUITOS LÓGICOS 32
3.1 Introdução 32
3.2 Variáveis e Expressões na Álgebra de Boole 32

3.3 Postulados 32
3.4 Simplificação de Expressões Booleanas 34

4. DIAGRAMAS DE VEITCH-KARNAUGH 36
4.1 Método da Soma de Produtos 36
4.2 Equação da Soma de Produtos 36
4.3 Circuito Lógico 37
4.4 Tabela-Verdade para Mapa de Karnaugh 38
4.5 Mapas de Três Variáveis 39
4.6 Mapas de Quatro Variáveis 40
4.7 Pares, Quadros e Octetos 40
4.8 Quadros 41
4.9 O Octeto 41
4.10 Simplificações de Karnaugh 41
4.11 Sobrepondo Grupos 42
4.12 Enrolando o Mapa 43

5. CIRCUITOS COMBINACIONAIS (1ª PARTE) 44
5.1 Introdução 44
5.2 Projetos de Circuitos Combinacionais 44
5.3 Circuitos com 2 Variáveis 45
5.4 Circuitos com 3 Variáveis 46

6. CIRCUITOS COMBINACIONAIS (2ª PARTE) 48
6.1 Introdução 48
6.2 Códigos 48
6.3 Códigos BCD 8421 48
6.4 Codificadores e Decodificadores 49
6.5 Codificadores Decimais / Binários 50
6.6 Decodificadores Binários / Decimais 51

6.7 Decodificador para Display de 7 Segmentos 53
6.8 Circuitos Aritméticos 56
6.9 Meio Somador 56
6.10 Somador Completo 57
6.11 Meio Subtrator 60
6.12 Subtrator Completo 62
6.13 Somador / Subtrator Completo 64

7. FLIP – FLOP REGISTRADORES E CONTADORES 66
7.1 Introdução 66
7.2 Flip – Flops 66
7.3 Flip –Flop RS Básico 67
7.4 Flip – Flop RS com Entrada Clock 68
7.5 Flip – Flop JK 70
7.6 Flip – Flop JK com Entradas Preset e Clear 70
7.7 Flip – Flop JK Mestre-Escravo 71
7.8 Flip – Flop JK Mestre-Escravo com Entrada Preset e Clear 72
7.9 Flip – Flop Tipo T 73
7.10 Flip – Flop Tipo D 74
7.11 Registradores de Deslocamento 75
7.12 Conversor Série-Paralelo 75
7.13 Conversor Paralelo-Série 77
7.14 Contadores 78
7.15 Contadores Utilizados em Circuitos Temporizadores 85

8. CIRCUITOS MULTIPLEX E DEMULTIPLEX 87
8.1 Introdução 87
8.2 Multiplex 87
8.3 Projeto do Circuito de um Multiplex 88
8.4 Amplificação da Capacidade de um Sistema Multiplex 89

8.4 Amplificação da Capacidade de um Sistema Multiplex 89
8.5 Demultiplex 92
8.6 Projeto do Circuito de um Demultiplex 92
8.7 Ampliação da Capacidade de um Circuito Demultiplex 94
8.8 Multiplex e Demultiplex Utilizados na Transmissão de Dados 95

9. FAMÍLIAS DE CIRCUITOS LÓGICOS 97
9.1 Níveis de Tensão e de Corrente 97
9.2 Características Gerais e Parâmetros da Família TTL 98
9.3 Versões dos Circuitos TTL 99
9.4 Características Gerais e Parâmetros da Família CMOS 100
9.5 Circuitos Integrados CMOS 101
9.6 Circuitos Integrados TTL 102

10. ENSAIOS DE ELETRÔNICA DIGITAL 103
10.1 Ensaio 1 – Introdução a Eletrônica Digital 103
10.2 Ensaio 2 – Portas Lógicas 106
10.3 Ensaio 3 – Álgebra Booleana 109
10.4 Ensaio 4 – Circuitos Combinacionais 110
10.5 Ensaio 5 – Decodificador BCD 8421 para Display de 7 Segmentos 116
10.6 Ensaio 6 – Circuitos Aritméticos 117
10.7 Ensaio 7 – Flip-Flops 119
10.8 Ensaio 8 – Contadores Assíncronos 121
10.9 Ensaio 9 – Contadores Síncronos 124
10.10 Ensaio 10 – Multiplex / Demultiplex 127
10.11 Ensaio 11 – Registradores de Deslocamento 131
11. DIGITAL INTEGRATED CIRCUITS 133
11.1 Function Selection Chart 133
11.2 Functional Diagrams 136
11.3 Packages and Ordering Information 149

11.4 Comparisan Table for Integrated Circuits 150
11.5 TTL Series (Transistor-Transistor-Logic) 151
11.6 LSL-Series (Low-Speed Noise- Immune Logic) 171
REFERÊNCIAS BIBLIOGRÁFICAS

1 SISTEMA DE NUMERAÇÃO

1.1 Introdução

O homem, através dos tempos, sentiu a necessidade da utilização de sistemas numéricos.

Existem vários sistemas numéricos, dentre os quais se destacam: o *sistema decimal*, o *binário*, o *octal* e o *hexadecimal*.

O sistema decimal é utilizado por nós no dia-a-dia e é, sem dúvida, o mais importante dos sistemas numéricos. Trata-se de um sistema que possui dez algarismos, com os quais podemos formar qualquer número, através da lei de formação.

Os sistemas: binário, octal e hexadecimal são muito importantes na área de técnicas digitais e computação.

1.2 O Sistema Binário de Numeração

O sistema binário de numeração é um sistema no qual existem apenas dois algarismos:

- O algarismo 0 (zero), e
- O algarismo 1 (um).

1.3 Odômetro Decimal

Para entender como contar com números binários vamos ver como um odômetro (indicador de quilômetros de um carro) conta com números decimais. Quando o carro é novo, seu odômetro começa com:

0	0	0	0	0

Após um quilômetro a leitura se torna:

0	0	0	0	1

Quilômetros sucessivos produzem 00002, 00003 e assim por diante, até:

0	0	0	0	9

Algo familiar ocorre ao final do décimo quilômetro. Quando a roda das unidades comuta de 9 outra vez para 0, um pino nessa roda força a roda das dezenas a avançar de 1. É por isso que o número muda para:

0	0	0	1	0

A roda das unidades foi colocada em 0 e enviou um vai-um para a roda das dezenas. Chamemos esse processo familiar de zeragem e vai-um. As outras rodas de um odômetro também são zeradas e enviam-um. Por exemplo, após 999 quilômetro mostra:

0	0	9	9	9

O que o próximo quilômetro faz? A roda das unidades é zerada e envia-um, a roda das dezenas é zerada e envia-um, roda das centenas é zerada e envia-um, e a roda dos milhares avança de 1, para obter:

0	1	0	0	0

1.4 Odômetro Binário

Imagine um odômetro binário, um dispositivo cujas rodas tem somente dois dígito, 0 e 1. Quando cada roda comuta, ele mostra 0, depois 1, novamente 0, e o ciclo se repete. Um odômetro binário de quatro dígitos começa com:

0	0	0	0

Após um quilômetro ele indica:

0	0	0	1

O próximo quilômetro força a roda das unidades a zerar e enviar-um; assim os números mudam para:

0	0	1	0

O terceiro quilômetro resulta em:

0	0	1	1

Após quatro quilômetro, a roda das unidades zera e envia-um, a segunda roda zera e envia-um, e a terceira roda avança de 1:

0	1	0	0

A tabela mostra todos os números binários de 0000 a 1111, equivalentes aos decimais de 0 a 15.

Decimal	Binário	Decimal	Binário
0	0000	8	1000
1	0001	9	1001
2	0010	10	1010
3	0011	11	1011
4	0100	12	1100
5	0101	13	1101
6	0110	14	1110
7	0111	15	111

1.5 Conversão do Sistema Binário para o Sistema Decimal

Para explicar a conversão vamos utilizar um número decimal qualquer, por exemplo, o número 594. Este número significa:

$$5 \times 100 + 9 \times 10 + 4 \times 1 = 594$$

Centena dezena unidade

$$5 \times 10^2 + 9 \times 10^1 + 4 \times 10^0 = 594$$

Esquematicamente, temos:

100	10	1
5	9	4

$\rightarrow 5 \times 100 + 9 \times 10 + 4 \times 1 = 594$

10^2	10^1	10^0
5	9	4

$\rightarrow 5 \times 10^2 + 9 \times 10^1 + 4 \times 10^0 = 594$

Neste exemplo podemos notar que o algarismo menos significativo (4) multiplica a unidade (1 ou 100), o segundo algarismo (9) multiplica a dezena (10 ou 101) e o mais significativo (5) multiplica a centena (100 ou 102). A soma desses resultados irá representar o número.

Podemos notar ainda, que a regra básica de formação de um número consiste no somatório de cada algarismo correspondente multiplicado pela base (no exemplo o número dez) elevada por índice conforme o posicionamento do algarismo no número.

Vamos agora utilizar um número binário qualquer, por exemplo, o número 101. Pela tabela notamos que este número equivale ao número 5 no sistema decimal.

Utilizando o conceito básico de formação de um número, podemos obter a mesma equivalência, convertendo assim o número para o sistema decimal.

2^2	2^1	2^0
1	0	1

$$1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 5$$

$$\updownarrow \qquad \updownarrow \qquad \updownarrow$$

$$1 \times 4 + 0 \times 2 + 1 \times 1 = 5$$

O número 101 na base 2 é igual ao número 5 na base 10.
Daqui por diante, para melhor identificação do número, colocará como índice à base do sistema ao qual o número pertence. Assim sendo, para o exemplo podemos escrever:

$$101_2 = 5_{10}$$

Exercícios

1 – Converta o número 01110_2 em decimal.
2 – Converta o número 1010_2 em decimal.
3 – Converta o número 1100110001_2 em decimal.

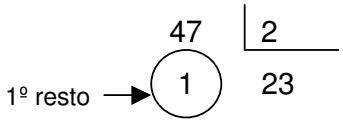
1.6 Conversão do Sistema Decimal para o Sistema Binário

Como vimos à necessidade da conversão sistema binário para decimal é evidente, pois, se tivermos um número grande no sistema binário, fica difícil perceber a quantidade que este representa. Transformando-se este número para decimal, o problema desaparece.

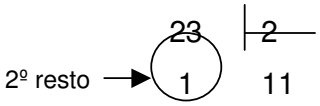
Veremos agora a transformação inversa, ou seja, a conversão de um número do sistema decimal para o sistema binário.

Para demonstrar o processo, vamos utilizar um número decimal qualquer, por exemplo, o número 47.

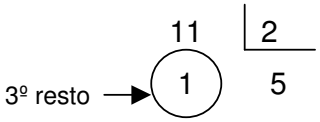
Dividindo o número 47 por 2, temos:



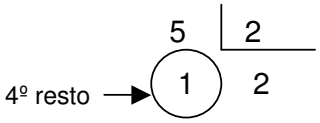
Dividindo agora 23 por 2, temos:



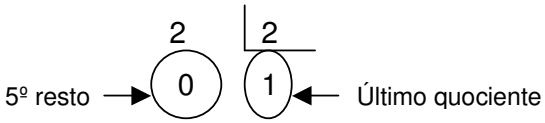
Dividindo agora 11 por 2, temos:



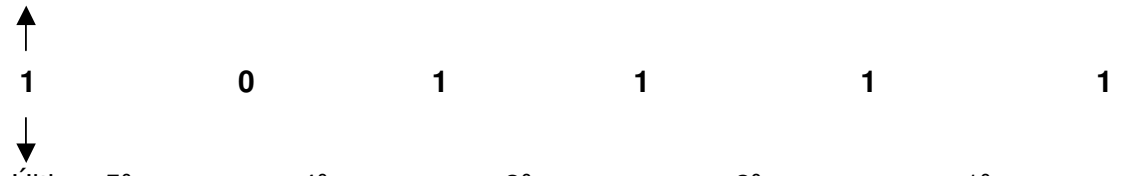
Dividindo agora 5 por 2, temos:



Dividindo agora 2 por 2, temos:



Método das divisões sucessivas consiste em efetuar-se sucessivas divisões pela base a ser convertida (no caso 2) até o último quociente possível. O número transformado será composto por este quociente (algarismo mais significativo) e, todos os restos, na ordem inversa às divisões. Dessa forma, temos:



$101111_2 = 47_{10}$

Na prática o bit menos significativo de um número binário recebe a notação de LSB (em inglês: Least Significant Bit) e o bit mais significativo de MSB (Most Significant Bit).

Exercícios:

1 – Converta o número 21_{10} em binário.
2 – Converta o número 552_{10} em binário.
3 – Converta o número 715_{10} em binário.

1.7 O Sistema Octal de Numeração

O sistema octal de numeração é um sistema no qual existem oito algarismos:

0, 1, 2, 3, 4, 5, 6, 7

Para representarmos a quantidade oito, agimos do mesmo modo, visto anteriormente, para números binários e decimais. Colocamos o algarismo 1 seguido do algarismo 0.

Atualmente, o sistema Octal praticamente é pouco utilizado no campo da Eletrônica Digital, tratando apenas de um sistema numérico intermediário dos sistemas binário e hexadecimal.

A tabela mostra a seqüência de numeração do sistema octal.

Decimal	Octal	Decimal	Octal
0	0	10	12
1	1	11	13
2	2	12	14
3	3	13	15
4	4	14	16
5	5	15	17
6	6	16	20
7	7	17	21
8	10	18	22
9	11	19	23

1.8 Conversão do Sistema Octal para o Sistema Decimal

Para convertermos um número octal em decimal, utilizamos o conceito básico de formação de um número, conforme já visto.

Vamos, por exemplo, converter o número 144_8 em decimal:

8^2	8^1	8^0
1	4	4

$1 \times 8^2 + 4 \times 8^1 + 4 \times 8^0 =$

$1 \times 64 + 4 \times 8 + 4 \times 1 = 64 + 32 + 4 = 100_{10}$

$\therefore 144_8 = 100_{10}$

Exercícios

1 – Converta o número 77_8 em decimal.

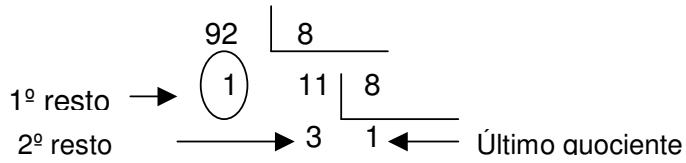
2 – Converta o número 100_8 em decimal.

3 – Converta o número 476_8 em decimal.

1.9 Conversão do Sistema Decimal para o Sistema Octal

O processo é análogo à conversão do sistema decimal para o binário, somente que neste caso, utilizaremos a divisão por 8, pois sendo o sistema octal, sua base é igual a 8.

Para exemplificar, vamos converter o número 92_{10} para o sistema octal:



$\therefore 92_{10} = 134_8$

Exercícios

1 – Converta o número 74_{10} em octal.

2 – Converta o número 512_{10} em octal.

3 – Converta o número 719_{10} em octal.

1.10 O Sistema Hexadecimal de Numeração

O sistema hexadecimal possui dezesseis algarismos, sendo sua base igual a 16. Os algarismos são assim enumerados:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Notamos que a letra A representa o algarismo A, que por sua vez representa a quantidade dez. A letra B representa o algarismo B, que representa a quantidade onze, e assim sucede-se até a letra F que representa a quantidade quinze.

Para representarmos a quantidade dezesseis, colocamos o algarismo 1 (um) seguido do algarismo 0 (zero).

Após esta introdução, podemos escrever a seqüência de numeração hexadecimal.

Decimal	Hexadecimal	Decimal	Hexadecimal
0	0	15	F
1	1	16	10
2	2	17	11
3	3	18	12
4	4	19	13
5	5	20	14
6	6	21	15
7	7	22	16
8	8	23	17
9	9	24	18
10	A	25	19
11	B	26	1A
12	C	27	1B
13	D	28	1C
14	E	29	1D

Este sistema é muito utilizado em microprocessadores e também no mapeamento de memórias de máquinas digitais com palavras 4, 8, 16 ou 32 bits.

1.11 Conversão do Sistema Hexadecimal para o Sistema Decimal

A regra de conversão é análoga à de outros sistemas, somente que neste caso, a base é 16. Como exemplo, vamos utilizar o número 3 F_{16} e convertê-lo em decimal:

16^1	16^0
3	F

$2 \times 16^1 + F \times 16^0 =$

Sendo $F_{16} = 15_{10}$, substituindo temos:

$3 \times 16^1 + 15 \times 16^0 = 3 \times 16 + 15 \times 1 = 63_{10}$

$\therefore 3\ F_{16} = 63_{10}$

Exercícios

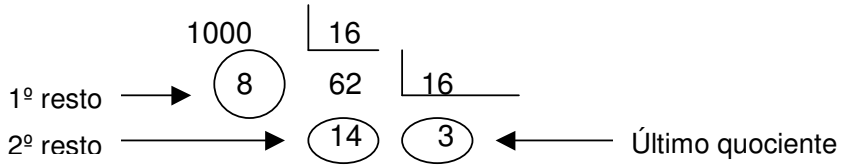
1 – Converta o número $1C3_{16}$ em decimal.

2 – Converta o número $238_{16.}$

3 – Converta o número 1 $FC9_{16}$ em decimal.

1.12 Conversão do Sistema Decimal para o Sistema Hexadecimal

Da mesma forma que nos casos anteriores, esta conversão se faz através de divisões sucessivas pela base do sistema a ser convertido. Para exemplificar vamos transformar o número 1000_{10} em hexadecimal:



Sendo $14_{10} = E_{16}$, temos: $3E8_{16}$

$\therefore$ $1000_{10} = 3E8_{16}$

Exercícios

1 – Converta o número 134_{10} para o sistema hexadecimal.

2 – Converta o número 384_{10} para o sistema hexadecimal.

3 –Converta o número 3882_{10} para o sistema hexadecimal.

2 FUNÇÕES E PORTAS LÓGICAS

2.1 Introdução

Em 1854, o matemático inglês **George Boole** (1815 – 1864), através da obra intitulada na **Investigation of the Laws of Thought,** apresentou um sistema matemático de análise lógica conhecido como **álgebra de Boole**.

No início da "era da eletrônica", todos os problemas eram resolvidos por sistemas analógicos, também conhecidos por sistemas lineares.

Apenas em 1938, o engenheiro americano **Claude Elwood Shannon** utilizou as teorias da álgebra de Boole para a solução de problemas de circuitos de telefonia com relés, tendo publicado um trabalho denominado **Symbolic Analysis of Relay and Switching**, praticamente introduzindo na área tecnológica o campo da eletrônica digital.

Esse ramo de eletrônica emprega em seus sistemas um pequeno grupo de circuitos básicos padronizados conhecidos como portas lógicas.

Através da utilização conveniente destas portas, podemos "implementar" todas as expressões geradas pela álgebra de Boole, que constituem a base dos projetos dos sistemas já referidos.

2.2 Funções Lógicas E, OU, NÃO, NE e NOU

Faremos, a seguir, o estudo das principais funções lógicas que na realidade derivam dos postulados da álgebra de Boole, sendo as variáveis e expressões envolvidas denominadas de booleanas.

Nas funções lógicas, temos apenas dois estados distintos:

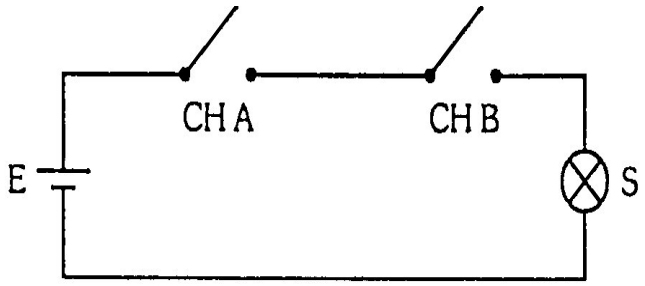
- ⇨ O estado **0** (zero) e
- ⇨ O estado **1** (um).

O estado 0 representará, Por exemplo: portão fechado, aparelho desligado, ausência de tensão, chave aberta, não, etc. O estado 1 representará, então: portão aberto, aparelho ligado, presença de tensão, chave fechada, sim, etc.

2.3 Função E ou AND

A função **E** é aquela que executa a multiplicação de 2 ou mais variáveis booleanas. É também conhecida como função **AND**, nome derivado do inglês. Sua representação algébrica para 2 variações é S = A **.** B, onde se lê S = A e B.

Para melhor compreensão, vamos utilizar e analisar o circuito representativo da função E visto na figura.



Convenções: chave aberta = 0 chave fechada = 1
Lâmpada apagada = 0 lâmpada acesa =1

Analisando as situações, concluímos que só teremos a lâmpada acesa quando as chaves A e B estiverem fechadas.

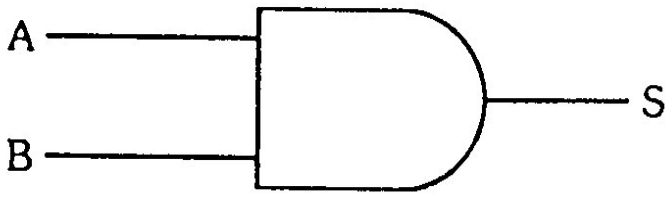
➢ **Tabela da Verdade de uma Função E ou AND**

Chamamos **Tabela da Verdade** um mapa onde colocamos todas as possíveis situações com seus respectivos resultados. Na tabela, iremos encontrar o modo como a função se comporta. A seguir, iremos apresentar a tabela da verdade de uma função E ou AND para 2 variáveis de entrada:

A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

- **Porta E ou AND**

A porta E é um circuito que executa a função E, sendo representada na prática, através do símbolo visto na figura.



Como já dissemos, a porta E executa a tabela da verdade da função E, ou seja, teremos a saída no estado 1 se, e somente se, as 2 entradas forem iguais a 1, e teremos a saída igual a 0 nos demais casos.

Notamos que a tabela da verdade mostra as 4 possíveis combinações das variáveis de entrada e seus respectivos resultados na saída.

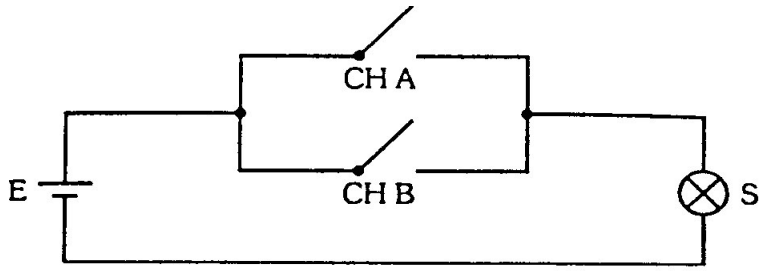
O número de situações possíveis a 2^N, onde N é o número de variáveis de entrada. Exemplo: N = 3 $\therefore 2^3 = 8$.

2.4 Função OU ou OR

A função OU é aquela que assume valor 1 quando uma ou mais variáveis da entrada forem iguais a 1 e assume valor 0 se, e somente se, todas as variáveis de entrada forem iguais a 0. Sua representação algébrica para 2 variáveis de entrada é S = A + B, onde se lê S = A ou B.

O termo OR, também utilizado, é derivado do inglês.

Para entendermos melhor a função OU, vamos representa-la através do circuito da figura.



Usaremos as mesmas convenções do circuito representativo da função E, visto anteriormente.

Notamos pelas situações que teremos a lâmpada ligada quando chA **ou** chB **ou** ambas as chaves estiverem ligadas.

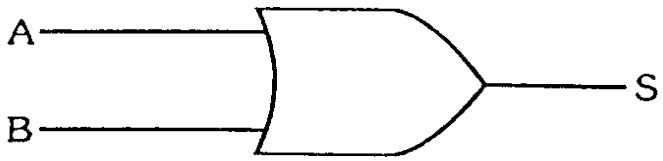
➢ **Tabela da Verdade da Função OU ou OR**

Nesta tabela da verdade, teremos todas as situações possíveis com os respectivos valores que a função OU assume. A tabela apresenta a tabela da verdade da função OU ou OR para 2 variáveis de entrada.

A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

- **Porta OU ou OR**

É a porta que executa a função OU. Representaremos a porta OU através do símbolo visto na figura.



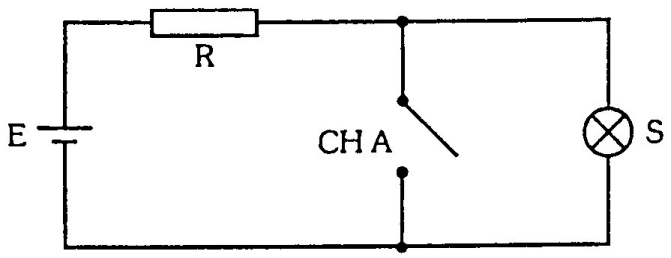
A porta OU executa a tabela da verdade de função OU, ou seja, teremos a saída igual a 1 quando uma ou mais variáveis de entrada forem iguais a 1 e 0 se, e somente se, todas as variáveis de entrada forem iguais a 0.

2.5 Função NÃO ou NOT

A função NÃO é aquela que inverte ou complementa o estado da variável, ou seja, se a variável estiver em 0, à saída vai para 1, e se estiver em 1, à saída vai para 0. É representada algebricamente da seguinte forma: S = $\overline{A}$ ou S = A', onde se lê **A barra** ou **NÃO A.**

Esta barra ou apóstrofo sobre a letra que representa a variável significa que esta sofre uma inversão. Também, podemos dizer que $\overline{A}$ significa a negação de A.

Para entendermos melhor a função NÃO vamos representá-la pelo circuito da figura. Analisaremos utilizando as mesmas convenções dos casos anteriores.



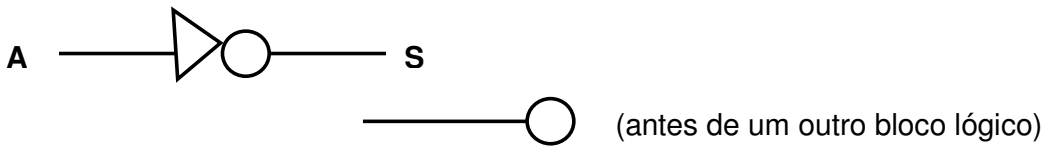
➢ **Tabela da Verdade da Função NÃO ou NOT**

A tabela apresenta casos possíveis da função NÃO.

A	S
0	1
1	0

➢ **Inversor**

O inversor é o bloco lógico que executa a função NÃO.
Suas representações simbólicas são vistas na figura.



2.6 Função NÃO E, NE ou NAND.

Como o próprio nome "NÃO E" diz: essa função é uma composição da função E com a função NÃO, ou seja, teremos a função E invertida. É representada algebricamente da seguinte forma:

$S = \overline{(A \cdot B)}$, onde o traço indica que temos a inversão do produto A.B.

➢ **Tabela da Verdade da Função NE ou NAND**

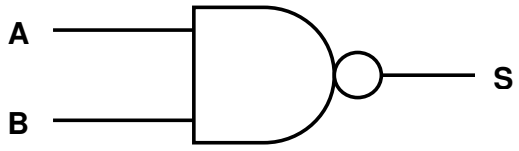
A tabela apresenta a função NE para 2 variáveis de entrada.

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

Pela tabela da verdade, podemos notar que esta função é o inverso da função E.

➢ **Porta NE ou NAND**

A porta NE é o bloco lógico que executa a função NE.
Sua representação simbólica é vista na figura.



2.7 Função NÃO OU, NOU ou NOR.

Analogamente à função NE, a função NOU é a composição da função NÃO com a função OU, ou seja, a função NOU será o inverso da função OU. É representada da seguinte forma:

S = ($\overline{A + B}$), onde o traço indica a inversão da soma booleana A + B.

➢ **Tabela da Verdade da Função NOU ou NOR**

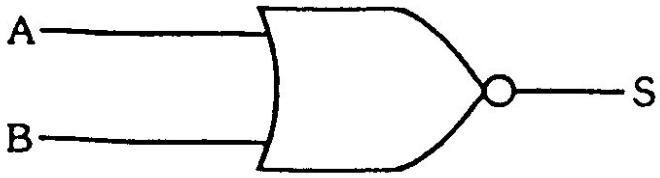
A tabela apresenta a função NOU para 2 variáveis de entrada.

A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

Podemos notar pela tabela da verdade que a função NOU representa a função Ou invertida.

- **Porta NOU ou NOR**

A porta NOU é o bloco lógico que executa a função NOU. Sua representação simbólica é vista na figura.



2.8 Bloco OU EXCLUSIVO

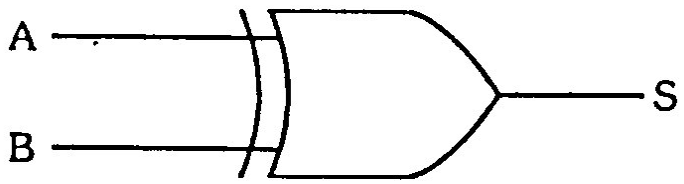
A função que ele executa, como o próprio nome diz, consiste em fornecer 1 à saída quando as variáveis de entrada forem diferentes entre si.

A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

Da tabela obtemos sua expressão característica:

$S = \overline{A} \cdot B + A \cdot \overline{B}$

A notação algébrica que representa a função OU Exclusivo é **S = A ⊕ B,** onde se lê A OU Exclusivo B, sendo S = A ⊕ B = $\overline{A}$ **.** B + A **.** $\overline{B}$. O circuito OU Exclusivo pode ser representado também pelo símbolo visto na figura.



Uma importante observação é que, ao contrário de outros blocos lógicos básicos, o circuito OU Exclusivo só pode ter 2 variáveis de entrada, fato este devido à sua definição básica. O circuito OU Exclusivo também é conhecido como Exclusive OR (EXOR), termo derivado do inglês.

2.9 Bloco COINCIDÊNCIA

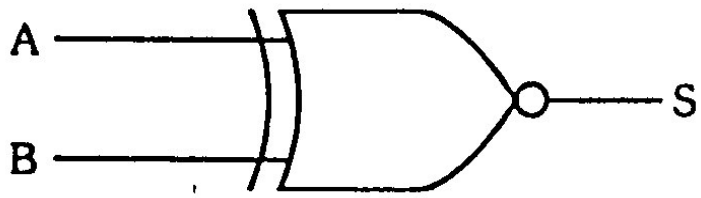
A função que ele executa, como seu próprio nome diz, é a de fornecer 1 à saída quando houver uma coincidência nos valores das variáveis de entrada.

Vamos, agora, montar sua tabela da verdade:

A	B	S
0	0	1
0	1	0
1	0	0
1	1	1

A tabela gera a expressão S = $\overline{A}$ **.** $\overline{B}$ + A.B.

A notação algébrica que representa a função Coincidência é S = A ⊙ B, onde se lê A Coincidência B, sendo S = A **.** B + $\overline{A}$ **.** $\overline{B}$. O símbolo do circuito Coincidência é visto na figura abaixo:



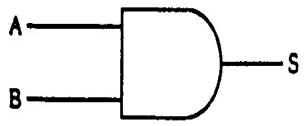
Se compararmos as tabelas da verdade dos blocos OU Exclusivo e Coincidência, iremos concluir que estes são complementares, ou seja, teremos a saída de um invertido em relação à saída do outro. Assim sendo, podemos escrever:

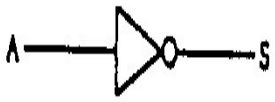
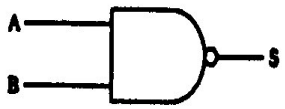
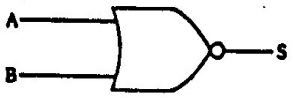
$$A \oplus B = \overline{A \odot B}$$

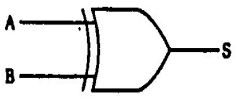
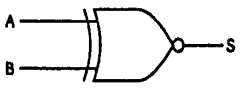
O bloco Coincidência é também denominado de NOU Exclusivo e do inglês Exclusive NOR.

Da mesma forma que o OU Exclusivo, o bloco Coincidência é definido apenas para 2 variáveis de entrada.

2.10 Quadro Resumo

BLOCOS LÓGICOS BÁSICOS				
Porta	**Símbolo Usual**	**Tabela da verdade**	**Função Lógica**	**Expressão**
E AND		A B S 0 0 0 0 1 0 1 0 0 1 1 1	**Função E:** Assume 1 quando todas as variáveis forem 1 e 0 nos outros casos.	S= AB
OU OR		A B S 0 0 0 0 1 1 1 0 1 1 1 1	**Função OU:** Assume 0 quando todas as variáveis forem 0 e 1 nos outros casos.	S=A+B

NÃO NOT INVERSOR		A | S 0 | 1 1 | 0	**Função NÃO:** Inverte a variável aplicada à sua entrada.	$S = \overline{A}$
NE NAND		A B | S 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0	**Função NE:** Inverso da função E.	$S = \overline{AB}$
NOU NOR		A B | S 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0	**Função NOU:** Inverso da função OU.	$S = \overline{A + B}$

OU EXCLUSIVO EXCLUSIVE OR		A B | S 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0	**Função OU:** Exclusivo – assume 1 quando as variáveis assumirem valores diferentes entre si	$S = \overline{A} \cdot B + A \cdot \overline{B}$ $S = A \oplus B$
NOU EXCLUSIVO EXCLUSIVE NOR COINCIDÊNCIA		A B | S 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1	**Função Coincidência:** Assume 1 quando houver coincidência entre os valores das variáveis	$S = \overline{A} \cdot \overline{B} + A \cdot B$ $S = A \odot B$

3 ÁLGEBRA DE BOOLE E SIMPLIFICAÇÃO DE CIRCUITOS LÓGICOS

3.1 Introdução

No capítulo anterior, trabalhamos com os circuitos lógicos sem nos preocuparmos com simplificações. Na prática, porém, estes circuitos obtidos admitem geralmente simplificações.

Para entrarmos no estudo da simplificação dos circuitos lógicos, teremos que fazer um breve estudo da Álgebra de Boole, pois é através de seus postulados, propriedades, teoremas fundamentais e identidades que efetuamos as mencionadas simplificações, e, além disso, notamos que é na Álgebra de Boole que estão todos os fundamentos da Eletrônica Digital.

3.2 Variáveis e Expressões na Álgebra de Boole

Como vimos anteriormente, as variáveis booleanas são representadas através de letras, podendo assumir apenas dois valores distintos: 0 ou 1. Denominamos expressão booleana a sentença matemática composta de termos cujas variáveis são booleanas, da mesma forma, podendo assumir como resultado final 0 ou 1.

3.3 Postulados

A seguir, apresentaremos os postulados da complementação, da adição e da multiplicação da Álgebra de Boole, e suas respectivas identidades resultantes.

POSTULADOS

Complementação	Adição	Multiplicação
$A = 0 \rightarrow \overline{A} = 1$ $A = 1 \rightarrow \overline{A} = 0$	$0 + 0 = 0$ $0 + 1 = 1$ $1 + 0 = 1$ $1 + 1 = 1$	$0 \cdot 0 = 0$ $0 \cdot 1 = 0$ $1 \cdot 0 = 0$ $1 \cdot 1 = 1$

IDENTIDADES

Complementação	Adição	Multiplicação
$\overline{\overline{A}} = A$	$A + 0 = A$ $A + 1 = 1$ $A + A = A$ $A + \overline{A} = 1$	$A \cdot 0 = 0$ $A \cdot 1 = A$ $A \cdot A = A$ $A \cdot \overline{A} = 0$

PROPRIEDADES

Comutativa:	$A + B = B + A$ $A \cdot B = B \cdot A$
Associativa:	$A + (B + C) = (A + B) + C = A + B + C$ $A \cdot (B \cdot C) = (A \cdot B) \cdot C = A \cdot B \cdot C$
Distributiva:	$A \cdot (B + C) = A \cdot B + A \cdot C$

TEOREMAS DE MORGAN

$$\overline{(A \cdot B)} = \overline{A} + \overline{B}$$

$$\overline{(A + B)} = \overline{A} \cdot \overline{B}$$

IDENTIDADES AUXILIARES

$$A + A \cdot B = A$$

$$A + \overline{A} \cdot B = A + B$$

$$(A + B) \cdot (A + C) = A + B \cdot C$$

3.4 Simplificação de Expressão Booleanas

Utilizando o conceito da Álgebra de Boole, podemos simplificar expressões e conseqüentemente circuitos.

Para efetuarmos estas simplificações, existem, basicamente, dois processos. O primeiro deles é a simplificação através da Álgebra de Boole; o segundo é a utilização dos mapas de Veitch Karnaugh.

Para elucidar, vamos utilizar, por exemplo, a expressão:

$$S = ABC + A\overline{C} + \overline{AB}$$

Vamos simplificá-la, utilizando a Álgebra de Boole. Primeiramente, vamos evidenciar o termo A:

$$S = A\left(BC + \overline{C} + \overline{B} \right)$$

Agora, aplicando a propriedade associativa, temos:

$$S = A\left(BC + (C + B) \right)$$

Aplicando a identidade X = X, temos:

$$S = A\left(BC + (C + B) \right)$$

Aplicando o teorema de De Morgan, temos:

$$S = \left(BC + \left(\overline{BC} \right) \right) A$$

Chamando BC de Y, logo $\overline{(\overline{BC})} = \overline{Y}$, temos, então:

$$S = A\ (Y+\overline{Y})$$

Como $Y + \overline{Y} = 1$, logo: $S = A \cdot 1 = A \therefore S = A$

Esta expressão mostra a importância da simplificação e a conseqüente minimização do circuito, pois os resultados são idênticos aos valores assumidos pela variável A, assim sendo, todo o circuito pode ser substituído por um único fio ligado à variável A.

Como um outro exemplo, vamos simplificar a expressão:

$$S = \overline{A}\ \overline{B}\ \overline{C} + \overline{A}\ B\ \overline{C} + A\ \overline{B}\ C$$

Tirando $\overline{A} \cdot \overline{C}$ em evidência nos dois primeiros termos, temos:

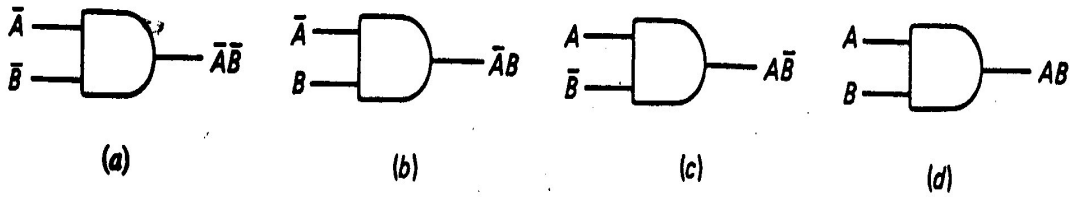
$$S = \overline{A} \cdot \overline{C} \cdot (\overline{B} + B) + A\overline{B}C$$

Aplicando a identidade: $B + \overline{B} = 1$, temos:

4 DIAGRAMAS DE VEITCH-KARNAUGH

4.1 Método da Soma de Produtos

A figura abaixo mostra as quatro maneiras possíveis de se fazer à operação AND de dois sinais de entrada que estão na forma complementada e não-complementada. Essas saídas são chamadas produtos fundamentais.



A tabela lista cada produto fundamental ao lado das condições de entrada para a produção de uma saída alta.

A	B	Produto Fundamental
0	0	$\bar{A}\ \bar{B}$
0	1	$\bar{A}B$
1	0	$A\ \bar{B}$
1	1	AB

4.2 Equação da Soma de Produtos

Aqui está como obter a solução de produtos dada uma tabela-verdade:

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Tabela –verdade de projeto

O que você tem a fazer é localizar cada saída na tabela-verdade e colocar por escrito o produto fundamental. Por exemplo, a primeira saída 1 aparece para uma entrada de A = 0, B = 1 e C = 1. O produto fundamental correspondente é ABC. A próxima saída 1 aparece para A = 1, B = 0 e C = 1. O produto fundamental correspondente é ABC. Continuando dessa forma, você pode identificar todos os produtos fundamentais, como mostrado na Tabela:

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	$1 \rightarrow \overline{A}BC$
1	0	0	0
1	0	1	$1 \rightarrow A\,\overline{B}C$
1	1	0	$1 \rightarrow AB\,\overline{C}$
1	1	1	$1 \rightarrow ABC$

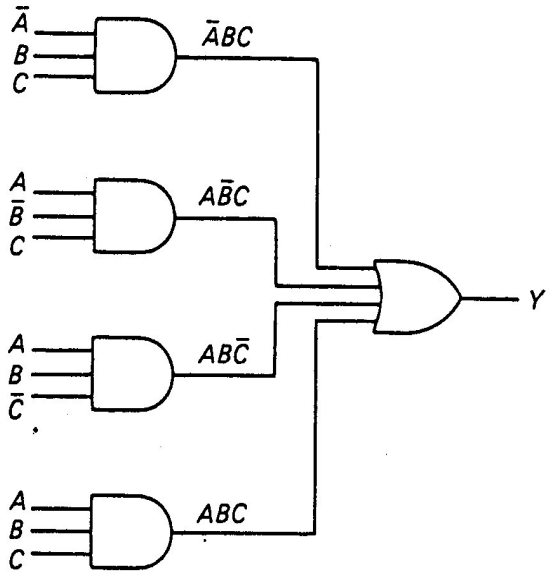
Produtos fundamentais

Para obter a equação da soma de produtos, você tem de fazer a operação OR dos produtos fundamentais da Tabela.

$$Y = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

4.3 Circuito Lógico

Após ter obtido uma equação de soma de produtos, você pode deduzir o correspondente circuito lógico desenhando uma rede AND-OR:

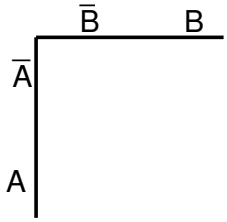


4.4 Tabela-verdade para Mapa de Karnaugh

Um mapa de Karnaugh é uma exposição visual dos produtos fundamentais necessários para uma solução de soma de produtos. Por exemplo, aqui está como converter a Tabela em um mapa de Karnaugh.

A	B	Y
0	0	0
0	1	0
1	0	1
1	1	1

Comece desenhando a figura:



Observe as variáveis e complementos: a coluna vertical tem $\overline{A}$ seguido por A, e a linha horizontal tem $\overline{B}$ seguido por B. agora, procure por saídas iguais a 1 na Tabela.

Indique esse produto fundamental no mapa da Karnaugh, como mostrado:

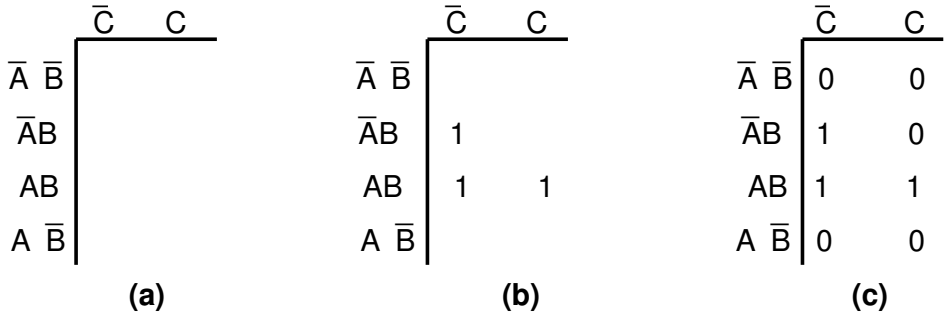
	$\overline{B}$	B
$\overline{A}$		
A	1	1

O passo final no desenho do mapa de Karnaugh é inserir os zeros nos espaços remanescentes:

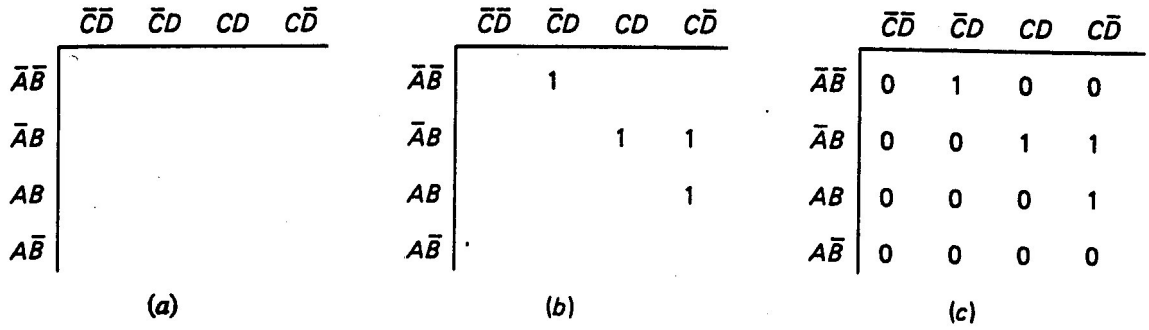
	$\overline{B}$	B
$\overline{A}$	0	0
A	1	1

4.5 Mapas de Três Variáveis

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

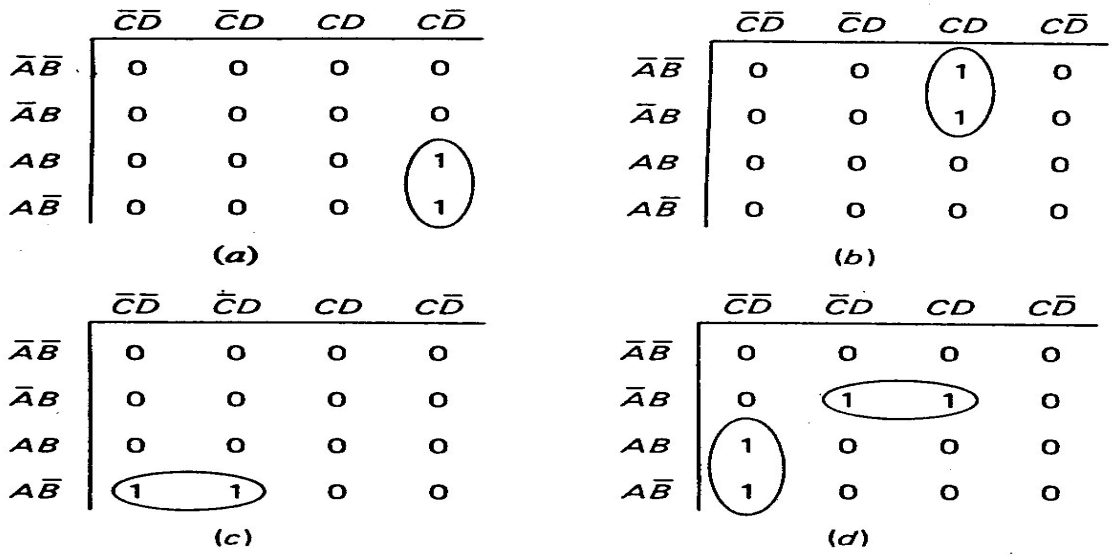


4.6 Mapas de Quatro Variáveis



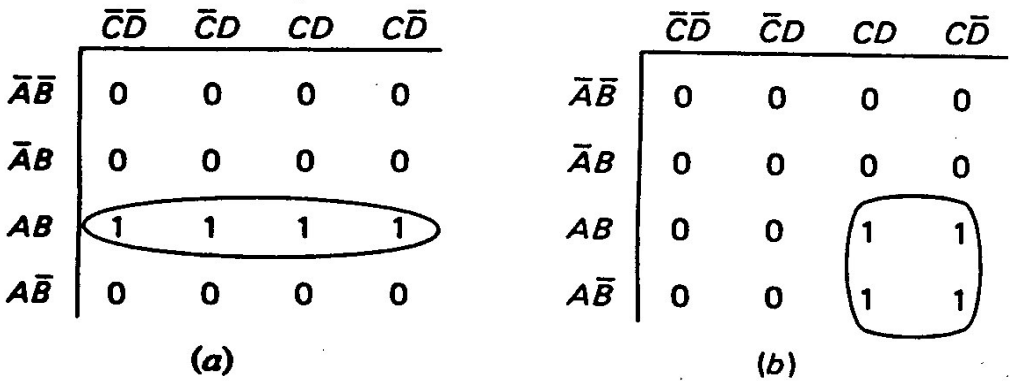
4.7 Pares, Quadras e Octetos.

O mapa contém um par de uns que são adjacentes.



4.8 Quadras

Uma quadra é um grupo de quatro uns, que são adjacentes horizontalmente ou verticalmente. Os uns podem estar lado a lado.



	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	0	0
AB	1	1	1	1
$A\bar{B}$	0	0	0	0

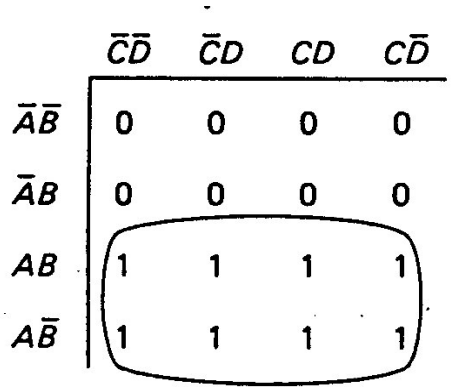
(a)

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	0	0
AB	0	0	1	1
$A\bar{B}$	0	0	1	1

(b)

4.9 O Octeto

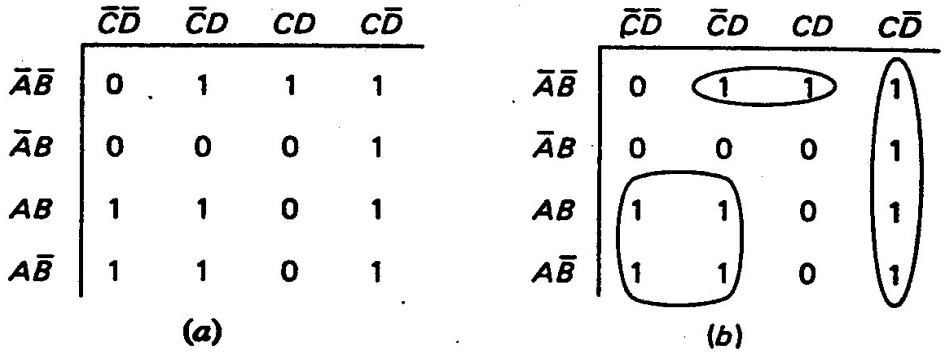
Além de pares e quadras, há mais um grupo de uns adjacentes para procurar: o octeto. Este é um grupo de oito uns.



	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	0	0
AB	1	1	1	1
$A\bar{B}$	1	1	1	1

4.10 Simplificações de Karnaugh

Como você sabe, um par elimina uma variável e seu complemento, uma quadra elimina duas variáveis e seus complementos, e um octeto elimina três variáveis e seus complementos. Por causa disso, após desenhar um mapa de Karnaugh, circunde primeiro os octetos, em segundo as quadras e, por último, os pares. Dessa forma consegue-se a maior simplificação.

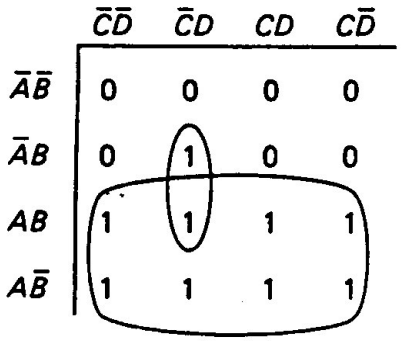


O par representa o produto simplificado A B D, a quadra inferior representa A C e a quadra da direita representa C D. Fazendo a operação OR desses produtos simplificados, obtemos a equação booleana correspondente ao mapa de Karnaugh completo:

$$Y = \overline{A}\ \overline{B}\ D + A\overline{C} + C\overline{D}$$

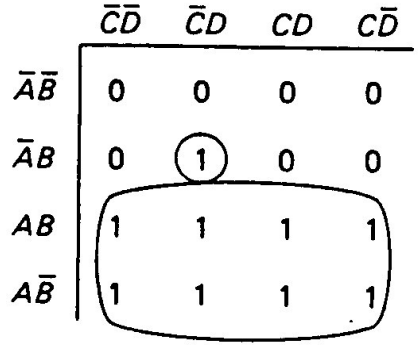
4.11 Sobrepondo Grupos

Você pode usar o mesmo 1 mais de uma vez.



$$Y = A + \overline{B}CD$$

É válido circundar os uns como mostrado na figura, mas o 1 isolado resulta em uma equação mais complicada:

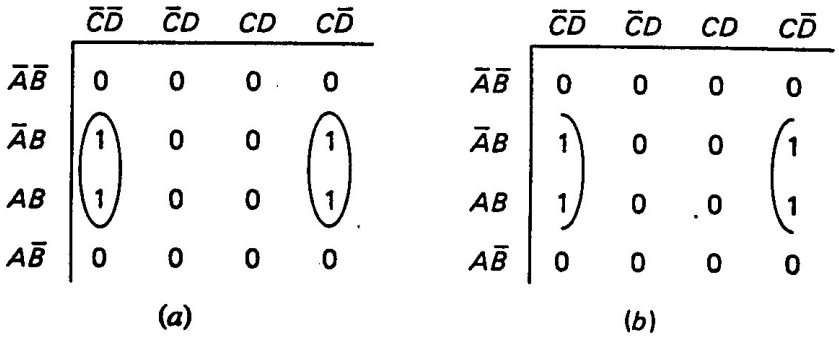


$$Y = A + \overline{A}BC\overline{D}$$

Por isto sobreponha grupos sempre que for possível. Isto é, use os uns mais de uma vez para obter os grupos maiores que você puder.

4.12 Enrolando o Mapa

Outra coisa, a saber, é sobre o enrolamento. Olhe para a figura. Os pares resultam nesta equação:



(a) (b)

5 CIRCUITOS COMBINACIONAIS – 1ª PARTE

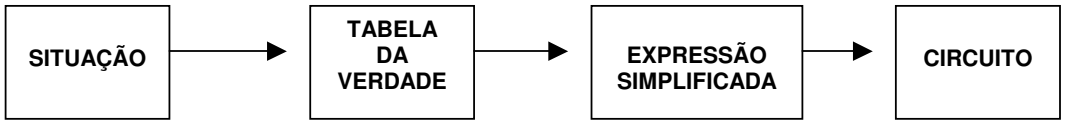
5.1 Introdução

Um dos capítulos importantes da Eletrônica Digital é o que trata dos circuitos combinacionais. É através do estudo destes que poderemos compreender o funcionamento de circuitos, tais como: somadores, subtratores, circuitos que executam prioridades, codificadores, decodificadores e outros muito utilizados na construção de computadores e em vários outros sistemas digitais.

O circuito combinacional é aquele em que a saída depende única e exclusivamente das combinações entre as variáveis de entrada.

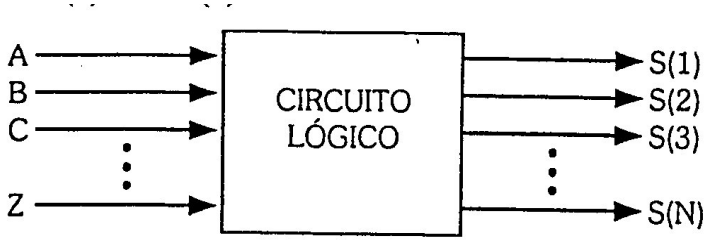
Podemos utilizar um circuito lógico combinacional para solucionar problemas em que necessitamos de uma resposta, quando acontecerem determinadas situações, representadas pelas variáveis de entrada. Para construirmos estes circuitos, necessitamos de suas expressões características que como vimos no capítulo anterior, são obtidas das tabelas da verdade que representam as situações já mencionadas.

A figura ilustra a seqüência do processo, onde, a partir da situação, obtemos a tabela da verdade e a partir desta, através das técnicas já conhecidas, a expressão simplificada e o circuito final.



5.2 Projetos de Circuitos Combinacionais

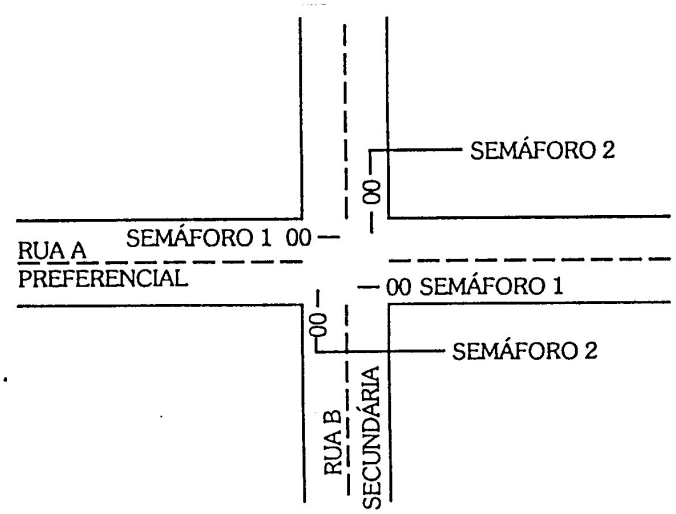
A figura mostra o esquema geral de um circuito combinacional composto pelas variáveis de entrada, o circuito propriamente dito e sua (s) saída (s).



Notamos que o circuito lógico pode possuir diversas variáveis de entrada e uma ou mais saídas conforme o caso do projeto.

5.3 Circuito com 2 Variáveis

A figura representa o cruzamento das ruas A e B. Neste cruzamento, queremos instalar um sistema automático para os semáforos, com as seguintes características:



1ª - Quando houver carros transitando somente na Rua B, o semáforo 2 deverá permanecer verde para que estas viaturas possam trafegar livremente;

2ª - Quando houver carros transitando somente na Rua A, o semáforo 1 deverá permanecer verde pelo mesmo motivo;

3ª - Quando houver carros transitando nas Ruas A e B, deveremos abrir o semáforo para a Rua A, pois é preferencial.

Primeiramente, vamos estabelecer as seguintes convenções:

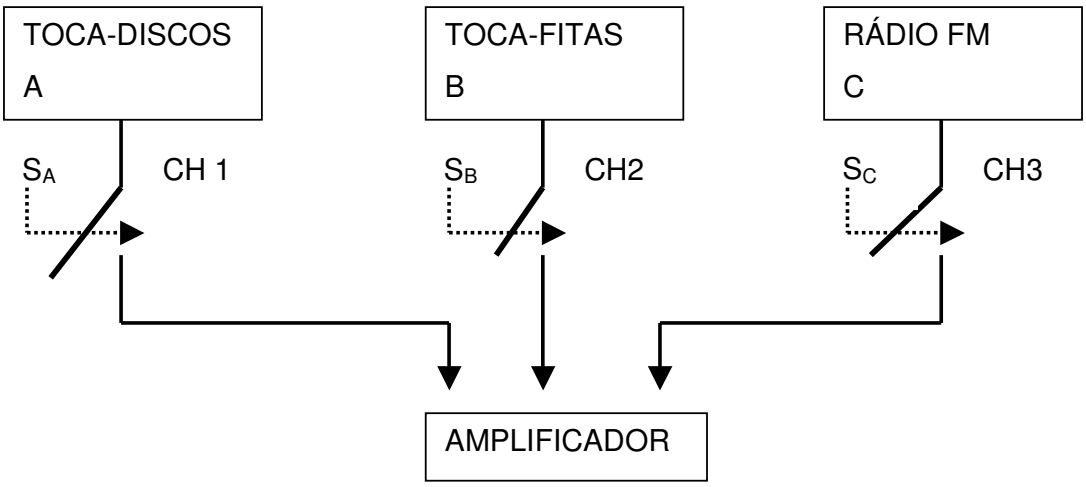
a) Existência de carro na Rua A: $A = 1$.
b) Não existência de carro na Rua A: $A = 0$ ou $\overline{A} = 1$.
c) Existência de carro na Rua B: $B = 1$.
d) Não existência de carro na Rua: $B = 0$ ou $\overline{B} = 1$.
e) Verde do sinal 1 acesso: $V_1 = 1$.
f) Verde do sinal 2 acesso: $V_2 = 1$.
g) Quando $V_1 = 1$ → vermelho do semáforo 1 apagando: $V_{m1} = 0$,
verde do semáforo 2 apagando: $V_2 = 0$
e vermelho do semáforo 2 acesso: $V_{m2} = 1$.
h) Quando $V_2 = 1$ → $V_1 = 0$, $V_{m2} = 0$ e $V_{m1} = 1$.

5.4 Circuito com 3 Variáveis

Deseja-se utilizar um amplificador para ligar três aparelhos: um toca-fitas, um toca-discos e um rádio FM. Vamos elaborar um circuito lógico que nos permitirá ligar os aparelhos, obedecendo às seguintes prioridades:

1ª prioridade: Toca-discos
2ª prioridade: Toca-fitas
3ª prioridade: Rádio FM

Isto significa que quando não ligarmos nem o toca-disco, nem o toca-fitas, o rádio FM, se ligado, será conectado à entrada do amplificador. Se ligarmos o toca-fitas, automaticamente o circuito irar conectá-lo à entrada do amplificador, pois possui prioridade sobre o rádio FM. Se, então, ligarmos o toca-disco, este será conectado ao amplificador, pois representa a 1ª prioridade. A partir disto, podemos montar o diagrama de blocos com as respectivas ligações:



Neste projeto, o circuito lógico receberá as informações das variáveis de entrada A,B e C, representando os aparelhos, e através das saídas S_A, S_B e S_C comutarão as chaves CH1, CH2 e CH3 para fazer a conexão conforme a situação requerida.

Convenções Utilizadas:

⇨ Variáveis de entrada (A,B e C): aparelho desligado = 0 e ligado = 1.
⇨ Saídas (S_A, S_B e S_C): S = 0 → chave aberta e S = 1 → chave fechada.

6 CIRCUITO COMBINACIONAIS - 2ª PARTE

6.1 Introdução

No capítulo anterior, vimos o processo de circuitos lógicos combinacionais utilizados na solução de problemas a partir de situações práticas de maneira geral. Neste capítulo, estudaremos outros, destinados principalmente a aplicações específicas, empregadas, sobretudo na arquitetura interna de circuitos integrados e, ainda, em sistemas digitais.

Entre os circuitos destinados a estas finalidades destacamos os codificadores, decodificadores e os circuitos aritméticos (meio somador, somador completo, meio subtrator e subtrator completo), que serão abordados a nível básico como projetos combinacionais, para melhor entendimento, sendo, entretanto encontrados na prática, disponíveis em circuitos integrados comerciais ou internos a sistemas mais complexos, tais como microprocessadores e circuitos integrados dedicados.

6.2 Códigos

São vários os códigos dentro do campo da Eletrônica Digital, existindo situações em que a utilização de um é vantajosa em relação a outro. Neste tópico, descrever os códigos mais conhecidos.

6.3 Códigos BCD 8421

Vamos iniciar explicando que no nome deste código, a sigla BCD representa as inicias de Binary Coded Decimal, que significa uma codificação do sistema decimal em binário. Os termos seguintes (8421) significam os valores dos algarismos num dado número binário, que conforme estudado no capítulo 1, representam respectivamente: 2^3, 2^2, 2^1 e 2^0.

A formação deste código é vista na tabela

DECIMAL	BCD 8421			
	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

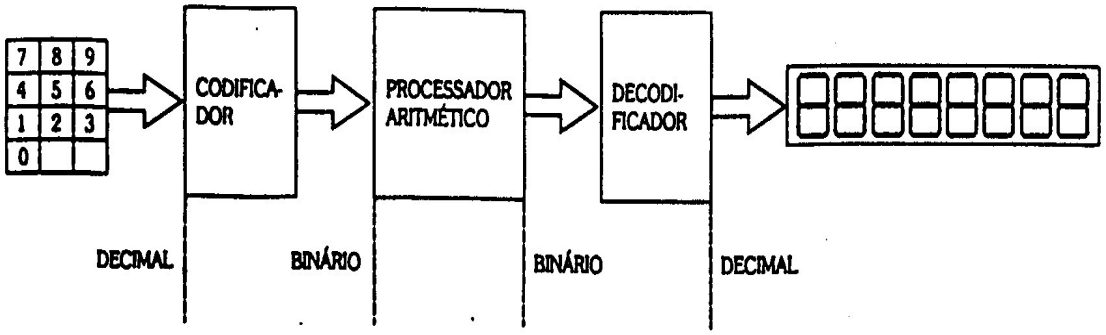
O número de bits de um código é o número de dígitos binários que este possui. Notamos, então, que o código BCD 8421 é um código de 4 bits e, ainda, que é válido de 0 a 9_{10}.

6.4 Codificadores e Decodificadores

Vamos, agora, tratar de circuitos que efetuam a passagens de um determinado código para outro. Primeiramente, vamos fazer uma análise do significado das palavras codificador e decodificador.

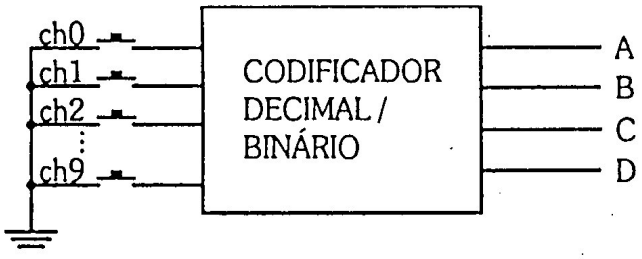
Chamamos de codificador o circuito combinacional que torna possível a passagem de um código conhecido para um desconhecido. Como exemplo, podemos citar o circuito inicial de uma calculadora que transforma uma entrada decimal, através do sistema de chaves de um teclado, em saída binária para que o circuito interno processe e faça a operação.

Chamamos de decodificador o circuito que faz o inverso, ou seja, passa um código desconhecido para um conhecido. No exemplo citado é o circuito que recebe o resultado da operação em binário e o transforma em saída decimal, na forma compatível para um mostrador digital apresentar os algarismos.



6.5 Codificador Decimal/Binário

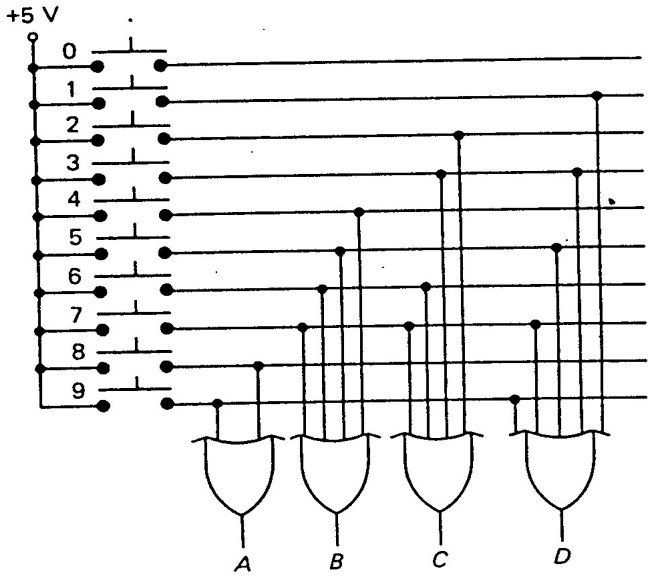
Vamos, neste item, elaborar um codificador para transformar um código decimal em binário (BCD 8421). A entrada do código decimal vai ser feita através de um conjunto de chaves numeradas de 0 a 9 e a saída por 4 fios, para fornecer um código binário de 4 bits, correspondente à chave acionada.



A seguir, vamos construir a tabela da verdade do codificador que relaciona cada chave de entrada decimal com a respectiva saída em binário:

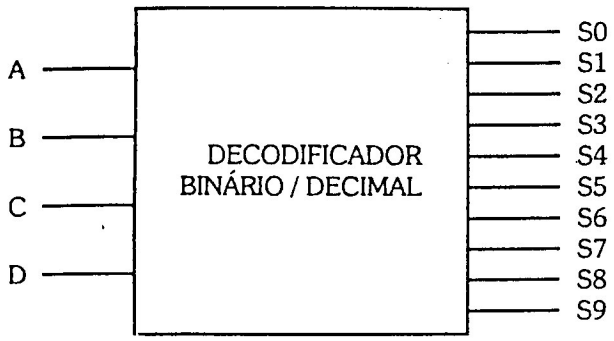
CHAVE	A	B	C	D
Ch0	0	0	0	0
Ch1	0	0	0	1
Ch2	0	0	1	0
Ch3	0	0	1	1
Ch4	0	1	0	0
Ch5	0	1	0	1
Ch6	0	1	1	0
Ch7	0	1	1	1
Ch8	1	0	0	0
Ch9	1	0	0	1

O circuito, assim constituído, é visto na figura.



6.6 Decodificador Binário/Decimal

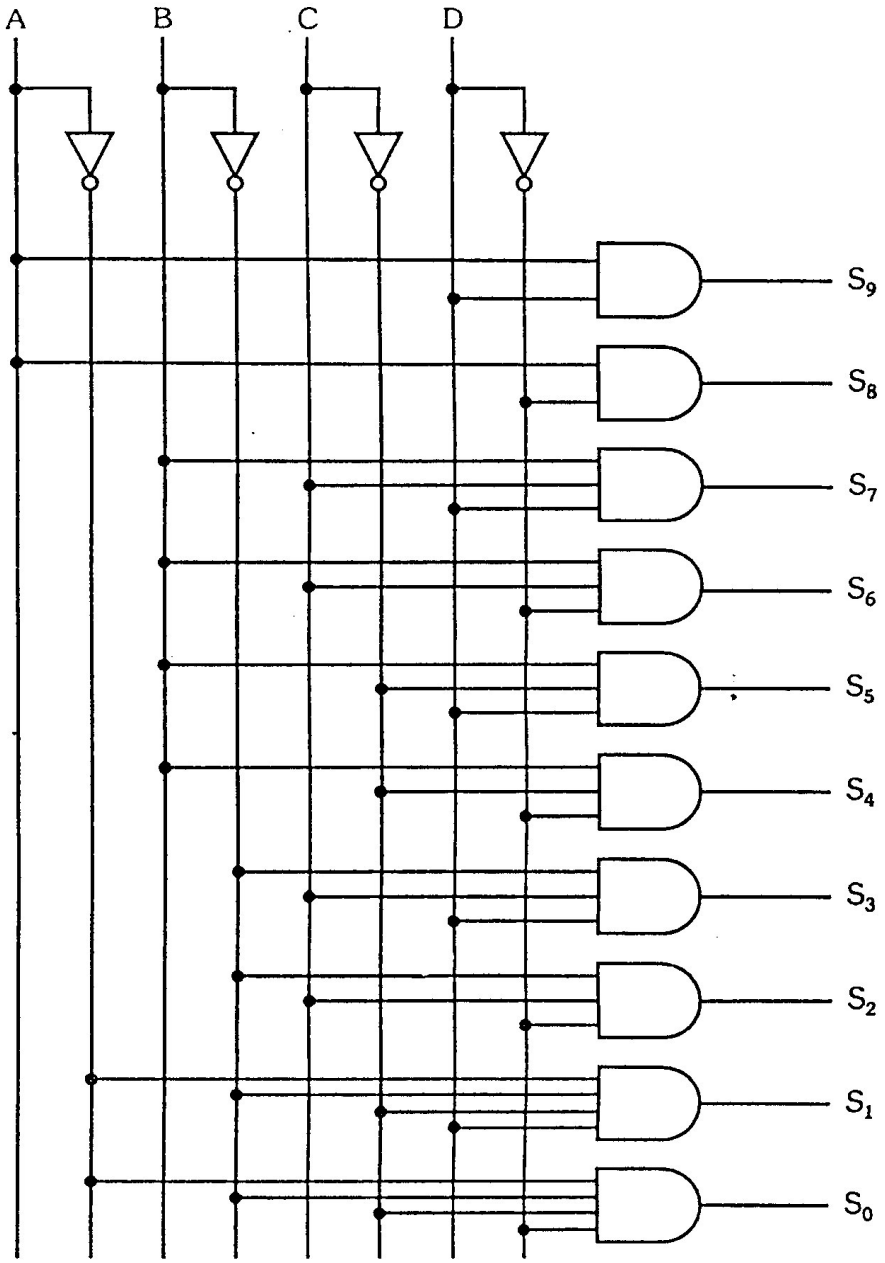
A estrutura geral deste decodificador é vista na figura.



Vamos montar a tabela da verdade do circuito no qual as entradas são bits do código BCD 8421 e as saídas são os respectivos bits do código decimal 98766543210.

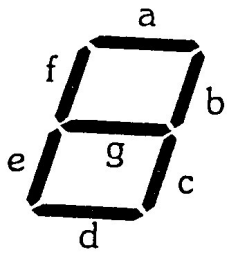
BCD 8421				**CÓDIGO 9876543210**									
A	**B**	**C**	**D**	**S9**	**S8**	**S7**	**S6**	**S5**	**S4**	**S3**	**S2**	**S1**	**S0**
0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0	0	0	0

A partir das expressões simplificadas, obtemos o circuito do decodificador que é visto na figura.

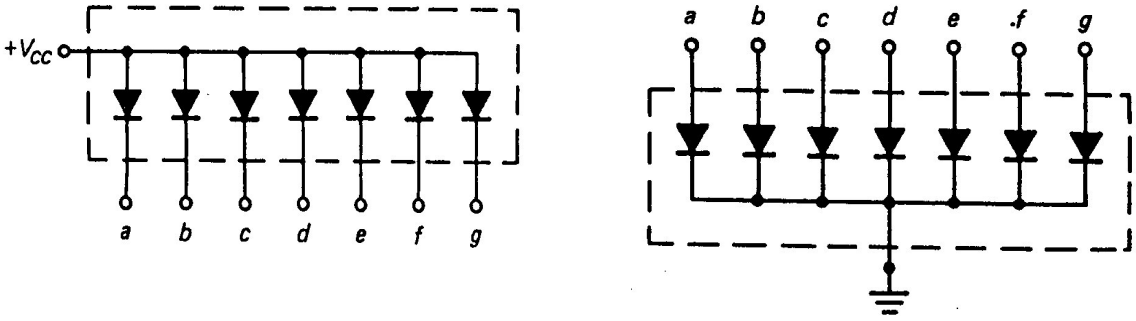


6.7 Decodificador para Display de 7 Segmentos

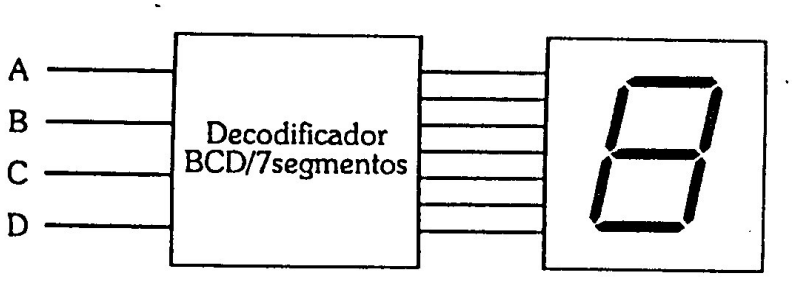
O **display de 7 segmentos** possibilita escrevermos números decimais de 0 a 9 e alguns outros símbolos que podem ser letras ou sinais. A figura representa uma unidade do display genérica, com a nomenclatura de identificação dos segmentos usual em manuais práticos.

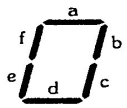
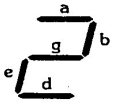
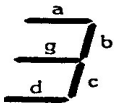
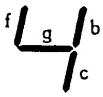
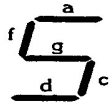
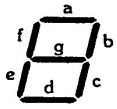
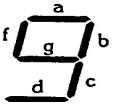


Entre as tecnologias de fabricação das unidades de display usaremos o mais comum que é o display a led, que possui cada segmento composto por um led, existindo um tipo denominado **catodo comum** e outro **anodo comum.**



Vamos a título de exemplo, elaborar um decodificador para a partir de um código binário (BCD 8421) escrever a seqüência de 0 a 9 em um display de 7 segmentos catodo comum. O esquema geral deste decodificador é visto na figura.



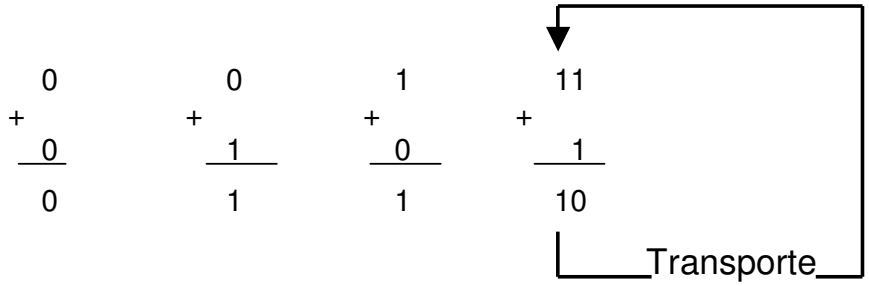
Characteres	Display	BCD 8421				Código para 7 Segmentos						
		A	B	C	D	a	b	c	d	e	f	g
0		0	0	0	0	1	1	1	1	1	1	0
1		0	0	0	1	0	1	1	0	0	0	0
2		0	0	1	0	1	1	0	1	1	0	1
3		0	0	1	1	1	1	1	1	0	0	1
4		0	1	0	0	0	1	1	0	0	1	1
5		0	1	0	1	1	0	1	1	0	1	1
6		0	1	1	0	1	0	1	1	1	1	1
7		0	1	1	1	1	1	1	0	0	0	0
8		1	0	0	0	1	1	1	1	1	1	1
9		1	0	0	1	1	1	1	1	0	1	1

6.8 Circuitos Aritméticos

Dentro do conjunto de circuitos combinacionais aplicados para finalidades específica nos sistemas digitais, destacam-se os circuitos aritméticos. São utilizados, principalmente, para construir a **ULA (Unidade Lógica Aritmética)** dos microprocessadores e, ainda, encontrados disponíveis em circuitos integrados comerciais. Neste tópico, abordamos os principais circuitos aritméticos e seus subsistemas derivados.

6.9 Meio Somador

Antes de iniciarmos o assunto, vamos relembrar alguns tópicos importantes da soma de 2 números binários:



Após essa breve introdução, vamos montar uma tabela da verdade da soma de 2 números binários de 1 algarismo:

A	B	S	Ts
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Ts → transporte de saída

(0 + 0 = 0 → Ts = 0)

(0 + 1 = 1 → Ts = 0)

(1 + 0 = 1 → Ts = 0)

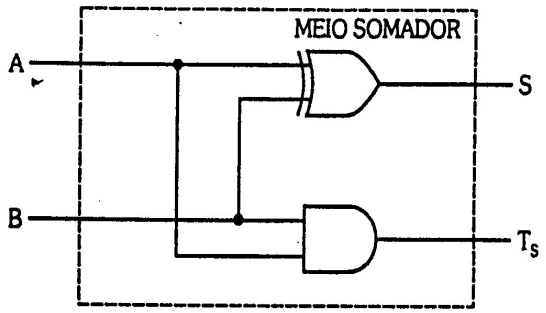
(1 + 1 = 0 → Ts = 1)

Representando cada número por 1 bit, podemos, então, montar um circuito que possui como entradas A e B, e como saída, a soma dos algarismos (S) e o respectivo transporte de saída (Ts). As expressões características do circuito, extraídas da tabela, são:

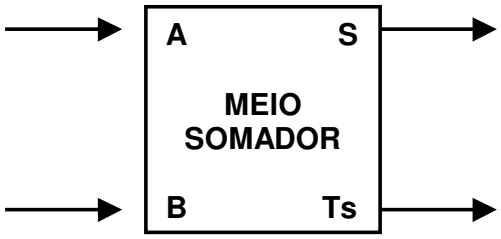
$$S = A \oplus B$$

$$Ts = AB$$

O circuito a partir destas expressões é visto na figura.

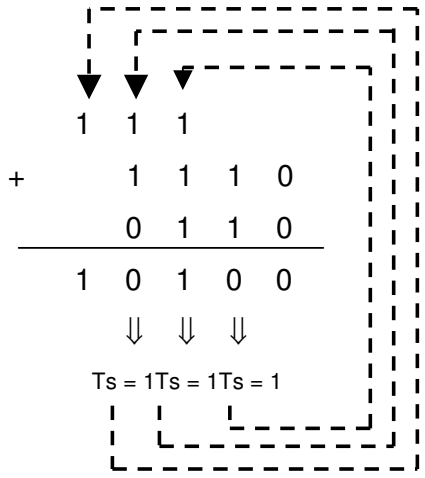


A representação em bloco deste circuito é vista na figura.



6.10 Somador Completo

O Meio Somador possibilita efetuar a soma de números binários com 1 algarismo. Para se fazer à soma de números binários de mais algarismos, esse circuito torna-se insuficiente, pois não possibilita a introdução do transporte de entrada proveniente da coluna anterior. Para melhor compreensão, vamos analisar o caso da soma: $1110_2 + 110_2$. Assim sendo, temos:



A coluna 1 tem como resultado um transporte de saída igual a 0. A coluna 2 tem como resultado 0 e um transporte de saída igual a 1. A coluna 3 tem um transporte de entrada igual a 1 (Ts da coluna anterior), possui resultado 1 e transporte de saída igual a 1. A coluna 4 tem transporte de entrada igual a 1, resultado 0 e transporte de saída 1. A coluna 5 possui apenas um transporte de entrada (Ts da coluna 4) e, obviamente, seu resultado será igual a 1.

Para fazermos a soma de 2 números binários de mais algarismos, basta somarmos coluna a coluna, levando em conta o transporte de entrada que nada mais é do que o Ts da coluna anterior.

O somador Completo é um circuito para efetuar a soma completa de uma coluna, considerando o transporte de entrada. Vamos, agora, montar a tabela da verdade deste circuito:

A	B	T_E	S	T_S
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

T_E → transporte de entrada

(0 + 0 + 0 = 0 → Ts = 0)

(0 + 0 + 1 = 1 → Ts = 0)

(0 + 1 + 0 = 1 → Ts = 0)

(0 + 1 + 1 = 0 → Ts = 1)

(1 + 0 + 0 = 1 → Ts = 0)

(1 + 0 + 1 = 0 → Ts = 1)

(1 + 1 + 0 = 0 → Ts = 1)

(1 + 1 + 1 = 1 → Ts = 1)

Vamos, então, escrever as expressões características, sem simplificação, de um Somador Completo:

$$S = \overline{A}\,\overline{B}T_E + \overline{A}B\overline{T_E} + A\overline{B}\overline{T_E} + ABT_E$$

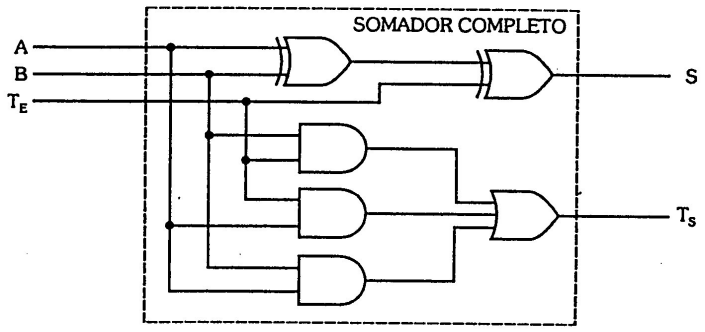
$$Ts = \overline{A}BT_E + A\overline{B}T_E + AB\overline{T_E} + ABT_E$$

Transpondo para diagramas de Veitch-Karnaugh, temos:

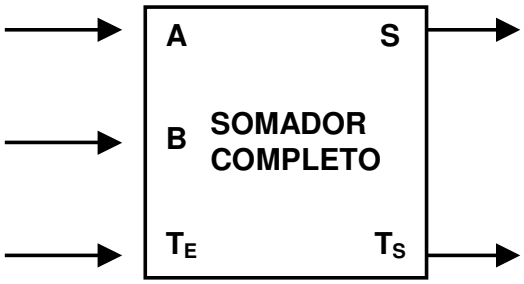
$$S = A \oplus B \oplus T_E$$

$$T_S = BT_E + AT_E + AB$$

Vamos, através das expressões, esquematizar o circuito Somador Completo:



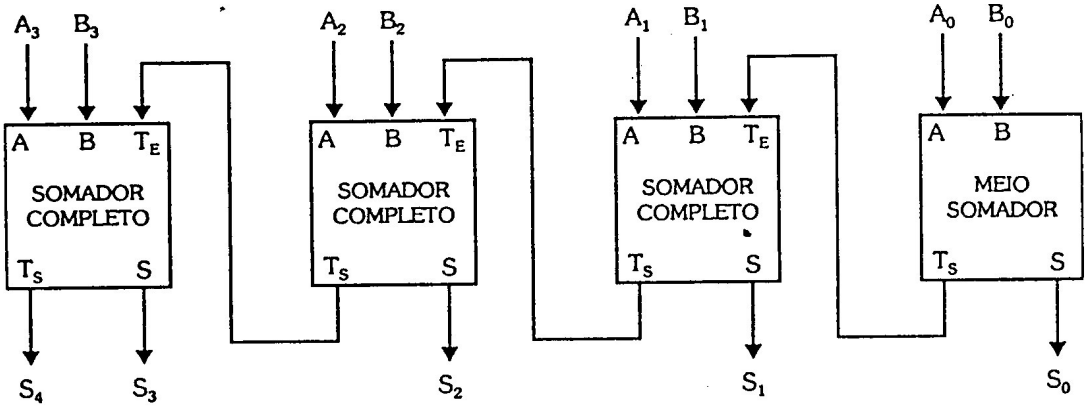
Da mesma forma, o circuito apresentado em bloco, é visto na figura.



Vamos, para exemplo de aplicação, montar um sistema em blocos que efetua soma de 2 números de 4 bits, conforme o esquema a seguir:

	A_3	A_2	A_1	A_0
+	B_3	B_2	B_1	B_0
S_4	S_3	S_2	S_1	S_0

Para efetuar a soma dos bits A_0 e B_0 dos números (1ª coluna), vamos utilizar um Meio Somador, pois não existe transporte de entrada, mas para as outras colunas utilizaremos Somadores Completos, pois necessitaremos considerar os transportes provenientes das colunas anteriores. O sistema montado é visto na figura.



6.11 Meio Subtrator

Antes de iniciarmos o assunto, vamos relembrar alguns tópicos importantes da subtração de números binários:

0 - 0 = 0

0 - 1 = 1 e transporta 1 (“empresta” 1)

1 - 0 = 1

1 - 1 = 0

Vamos montar a tabela da verdade de uma subtração de 2 números binários de 1 algarismo:

A	B	S	T_S
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

(0 – 0 = 0 → Ts = 0)

(0 – 1 = 1 → Ts = 1)

(1 – 0 = 1 → Ts = 0)

(1 – 1 = 0 → Ts = 0)

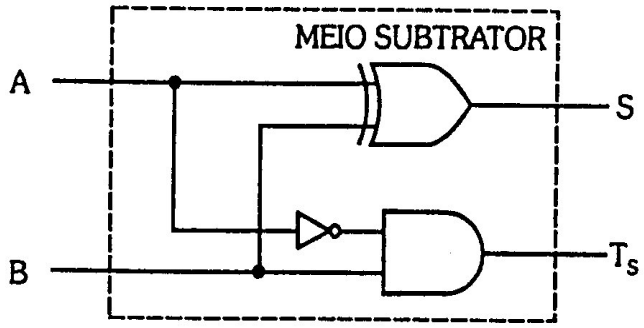
Representando cada número por 1 bit, podemos montar um circuito com as entradas A e B, e como saída, a subtração (S) e o transporte de saída (Ts).

As expressões características do circuito, extraídas da tabela, são:

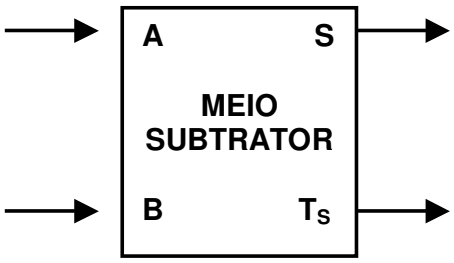
$S = A \oplus B$

$Ts = \overline{A}B$

O circuito a partir destas, é visto na figura.



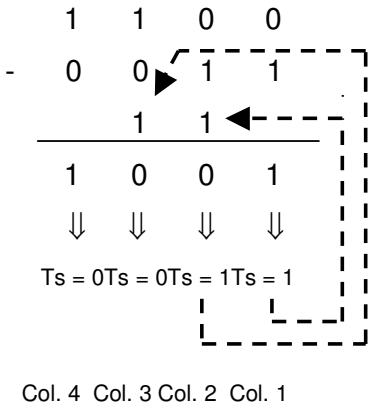
Em bloco, o circuito recebe a representação da figura.



6.12 Subtrator Completo

O Meio Subtrator possibilita-nos efetuar a subtração de números binários de 1 algarismo. Para se fazer uma subtração com números de mais algarismos, este circuito torna-se insuficiente, pois não possibilita a entrada do transporte (T_E), proveniente da coluna anterior.

Para compreendermos melhor, vamos analisar a subtração:
$1100_2 - 11_2$. Assim sendo, temos:



A coluna 1 tem como resultado de saída 1 e apresenta um transporte de saída igual a 1. a coluna 2 tem um transporte de entrada igual a 1 (Ts da coluna anterior), um resultado igual a 0 e Ts = 0. A coluna 4 tem: $T_E = 0$, resultado igual a 1 e Ts = 0.

Para fazermos a subtração de números binários de mais algarismos, basta subtrairmos coluna a coluna, levando em conta o transporte de entrada, que nada mais é do que o Ts da coluna anterior.

O Subtrator Completo é um circuito que efetua a subtração completa de uma coluna, ou seja, considera o transporte de entrada proveniente da coluna anterior. Vamos, agora, montar a tabela da verdade deste circuito:

A	B	T_E	S	Ts
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

As expressões características extraídas da tabela são:

$$S = \overline{A}\,\overline{B}T_E + \overline{A}B\overline{T}_E + A\overline{B}\overline{T}_E + ABT_E$$

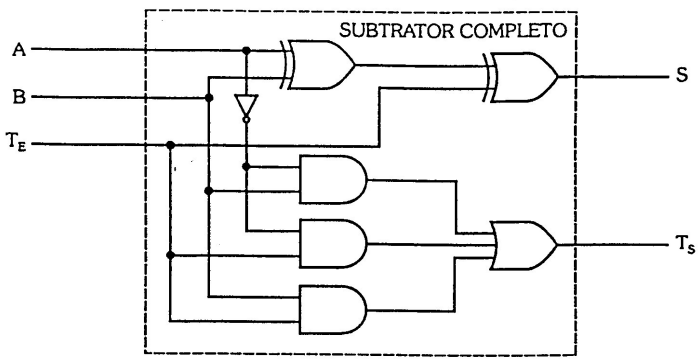
$$Ts = \overline{A}\,\overline{B}T_E + \overline{A}B\overline{T}_E + \overline{A}BT_E + ABT_E$$

Vamos simplificar estas expressões:

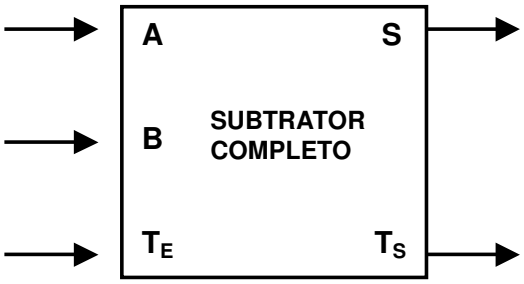
$$S = A \oplus B \oplus T_E$$

$$T_S = \overline{A}B + \overline{A}T_E + BT_E$$

O circuito derivado das expressões é visto na figura.



Em bloco, recebe a representação da figura.



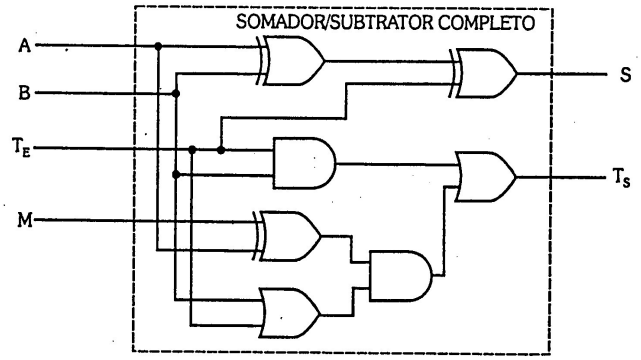
6.13 Somador / Subtrator Completo

Podemos esquematizar um circuito que efetue as duas operações. Para isso, vamos introduzir uma outra entrada que permanecendo em nível 0, faz o circuito efetuar uma soma completa, e permanecendo em nível 1, faz efetuar uma subtração completa.

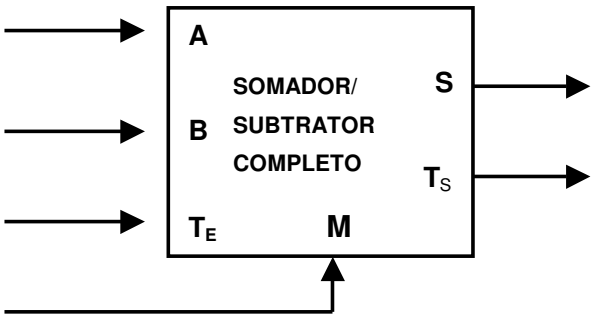
Vamos, agora, montar a tabela da verdade do circuito, sendo M a variável de controle (M = 0 $\rightarrow$ soma e M = 1 $\rightarrow$ subtração):

M	A	B	T_E	S	Ts	
0	0	0	0	0	0	Soma Completa (M = 0)
0	0	0	1	1	0	
0	0	1	0	1	0	
0	0	1	1	0	1	
0	1	0	0	1	0	
0	1	0	1	0	1	
0	1	1	0	0	1	
0	1	1	1	1	1	
1	0	0	0	0	0	Subtração Completa (M = 1)
1	0	0	1	1	1	
1	0	1	0	1	1	
1	0	1	1	0	1	
1	1	0	0	1	0	
1	1	0	1	0	0	
1	1	1	0	0	0	
1	1	1	1	1	1	

Vamos, então, esquematizar o circuito:



A figura mostra a representação deste circuito Somador/Subtrator Completo, em bloco:



7 FLIP-FLOP. REGISTRADORES E CONTADORES

7.1 Introdução

O campo da Eletrônica Digital é basicamente dividido em duas áreas: lógica combinacional e lógica seqüencial.

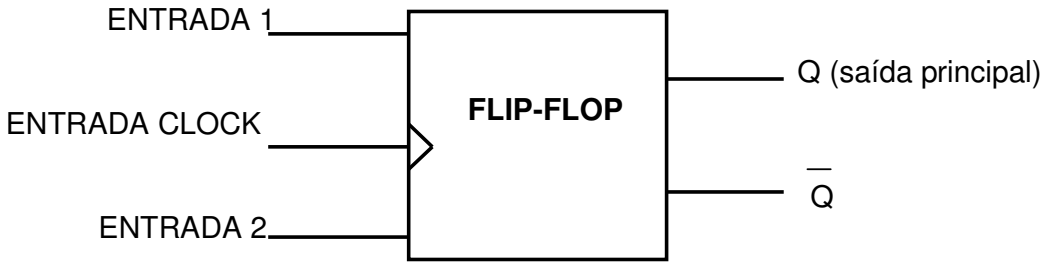
Os circuitos combinacionais, como vimos até aqui, apresentam as saídas, única e exclusivamente, dependentes das variáveis de entrada.

Os circuitos seqüenciais têm as saídas dependentes das variáveis de entrada e/ou de seus estados anteriores que permanecem armazenados, sendo, geralmente, sistemas pulsados, ou seja, operam sob o comando de uma seqüência de pulsos denominada **clock.**

Neste capítulo, trataremos do estudo dos flip-flops e de circuitos nos quais fazem o papel de elemento principal.

7.2 Flip-Flops

De forma geral, podemos representar o flip-flop como um bloco onde temos 2 saídas: Q e $\overline{Q}$, entradas para as variáveis e uma entrada de controle (clock). A saída Q será a principal do bloco. A figura ilustra um flip-flop genérico.



Este dispositivo possui basicamente dois estados de saída. Para o flip-flop assumir um destes estados é necessário que haja uma combinação das variáveis e do pulso de controle (clock). Após este pulso, o flip-flop permanecerá neste estado até a chegada de um novo pulso de clock e, então, de acordo com as variáveis de entrada, mudará ou não de estado.

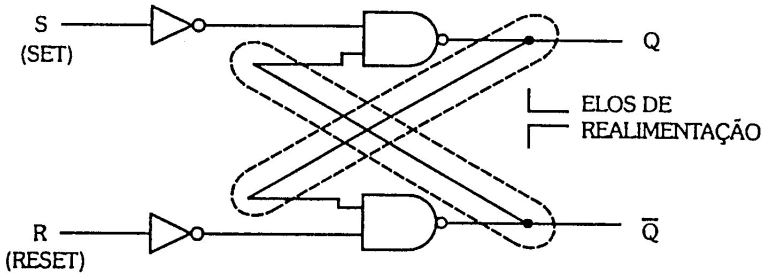
Os dois estados possíveis são:

1) $Q = 0 \rightarrow \overline{Q} = 1$
2) $Q = 1 \rightarrow \overline{Q} = 0$

Vamos, a seguir, analisar alguns circuitos de flip-flops e suas respectivas operações.

7.3 Flip-Flop RS Básico

Primeiramente, vamos analisar o flip-flop RS básico, construído a partir de portas NE e inversores, cujo circuito é visto na figura.



Notamos que estes elos de realimentação fazem com que as saídas sejam injetadas juntamente com as variáveis de entrada, ficando claro, então, que os estados que as saídas irão assumir dependerão de ambas.

Para analisarmos o comportamento do circuito, vamos construir a tabela da verdade, levando em consideração as 2 variáveis de entrada (S e R) e a saída Q anterior (Qa) à aplicação das entradas:

S	R	Qa	Qf	$\overline{Qf}$	
0	0	0	0	1	} fixa Qf = Qa
0	0	1	1	0	
0	1	0	0	1	} fixa Qf em 0
0	1	1	0	1	
1	0	0	1	0	} fixa Qf em 1
1	0	1	1	0	
1	1	0	1	1	} não permitido
1	1	1	1	1	

Podemos, então, resumir a tabela da verdade de um flip-flop RS básico:

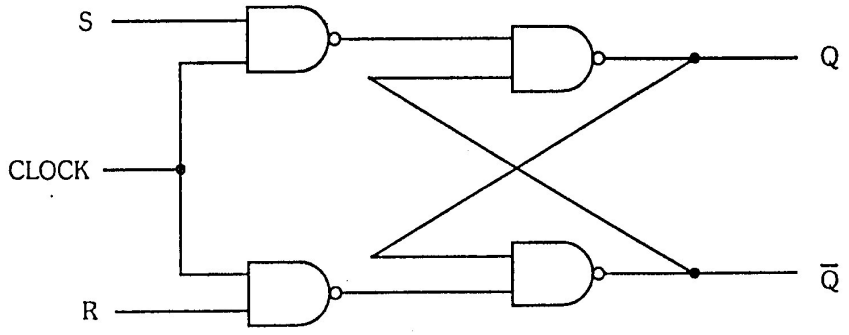
S	R	Qf
0	0	Qa
0	1	0
1	0	1
1	1	

A entrada S é denominada **Set**, pois quando acionada (nível 1), passa a saída para 1 (estabelece ou fixa 1), e a entrada R é denominada **Reset**, pois quando acionada (nível 1), passa a saída para 0 (recompõe ou zera o flip-flop). Estes termos são muito usuais na área de eletrônica digital, sendo provenientes do idioma inglês.

Este circuito irá mudar de estado apenas no instante em que mudam as variáveis de entrada. Veremos em seguida, como é o circuito de um flip-flop RS que tem sua mudança de estado controlada pela entrada de clock.

7.4 Flip-Flop RS com Entrada Clock

Para que o flip-flop RS básico seja controlado por uma seqüência de pulsos de clock, basta trocarmos os 2 inversores por portas NE, e ás outras entradas destas portas, injetarmos o clock. O circuito, com estas modificações é visto na figura.

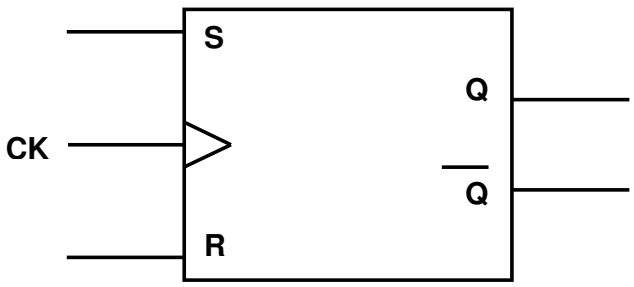


Neste circuito, quando a entrada do clock for igual a 0, o flip-flop irá permanecer no seu estado, mesmo que variem as entradas S e R. Isso pode ser confirmado pela análise do circuito, onde concluímos que para clock = 0, as saídas das portas NE de entrada serão sempre iguais a 1, independentemente dos valores assumidos por S e R.

Quando a entrada clock assumir valor 1, o circuito irá comportar-se como um flip-flop RS básico, pois as portas NE de entrada funcionarão como os inversores do circuito anteriormente visto. A tabela resume a operação deste flip-flop em função da entrada clock.

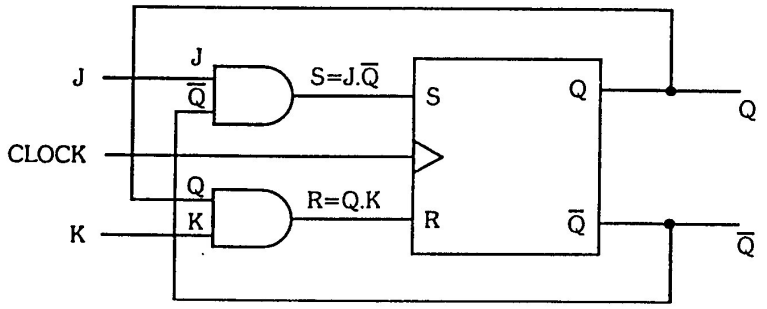
CK	Qf
0	QA
1	RS básico

De maneira geral, podemos concluir que o circuito irá funcionar quando a entrada clock assumir valor 1 e manterá travada esta saída quando a entrada clock passar para 0. O flip-flop pode ser representado pelo bloco visto na figura.



7.5 Flip-Flop JK

O flip-flop JK nada mais é que um flip-flop RS realimentado da maneira mostrada na figura.



A tabela simplificada resultante será:

J	K	Qf
0	0	Qa
0	1	0
1	0	1
1	1	$\overline{Qa}$

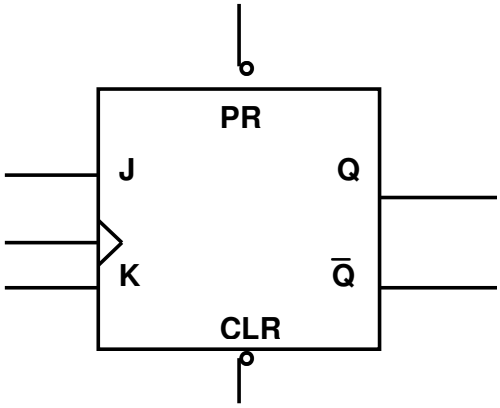
7.6 Flip-Flop JK com Entradas Preset e Clear

O flip-flop JK poderá assumir valores Q = 1 ou Q = 0 mediante a utilização das entradas **Preset (PR)** e **Clear (CLR)**.

A tabela resume a atuação das entradas Preset e Clear.

CLR	PR	Qf
0	0	Não permitido
0	1	0
1	0	1
1	1	Funcionamento normal

Podemos, para facilitar, utilizar um bloco representativo como o mostrado na figura.

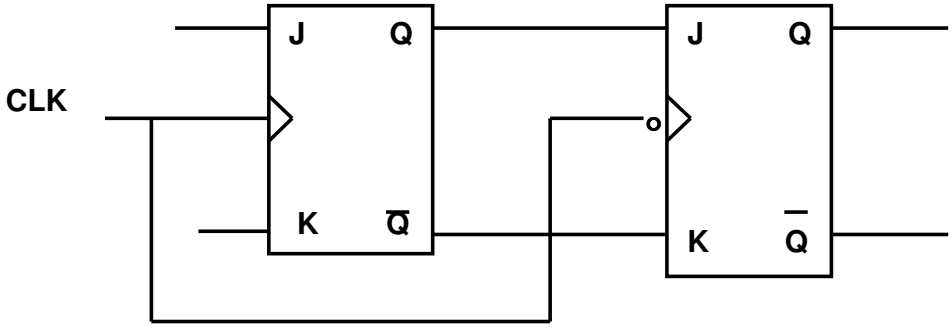


Os circuitos na simbologia do bloco indicam que as entradas Prest e Clear são ativas em 0, ou seja, funcionam respectivamente com nível 0 aplicado.

7.7 Flip-Flop JK Mestre-Escravo

O flip-flop JK apresenta uma característica indesejável. Quando o clock for igual a 1, teremos o circuito funcionando como sendo um circuito combinacional, pois haverá a passagem das entradas J, K e também da realimentação. Nessa situação, se houver uma mudança nas entradas J e K, o circuito apresentará uma nova saída, podendo alterar seu estado tantas vezes quantas alterarem os estados das entradas J e K.

Para resolver esse problema, foi criado o flip-flop **JK Mestre-Escravo (JK Master-Slave).**



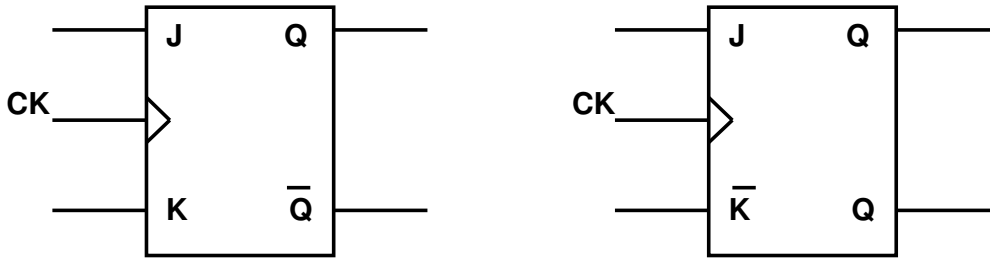
Flip-flop Mestre-Escravo.

A tabela resume a operação do flip-flop JK Mestre-Escravo:

J	K	Qf
0	0	Qa
0	1	0
1	0	1
1	1	$\overline{Qa}$

Notamos que esta tabela é idêntica à de um flip-flop JK básico, porém a saída Q irá assumir valores, conforme a situação das entradas JK, somente após a passagem do clock para 0. Assim sendo, o circuito é denominado JK Mestre-Escravo **sensível à descida de clock.** Para obter um circuito **sensível à subida de clock** basta colocarmos um inversor interno à entrada clock.

A figura mostra o bloco JK Mestre-Escravo e a simbologia para identificar o circuito sensível à descida de clock (a) e à subida de clock (b).

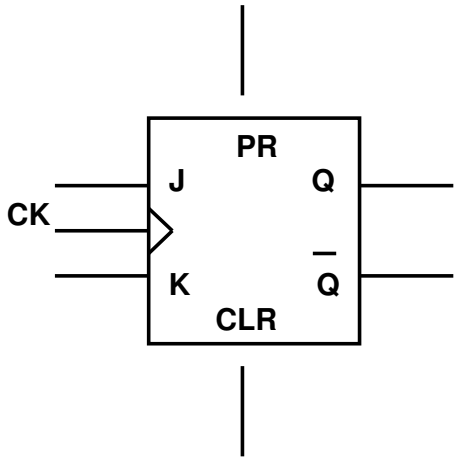


(a) (b)

7.8 Flip-Flop JK Mestre-Escravo com Entrada Preset e Clear

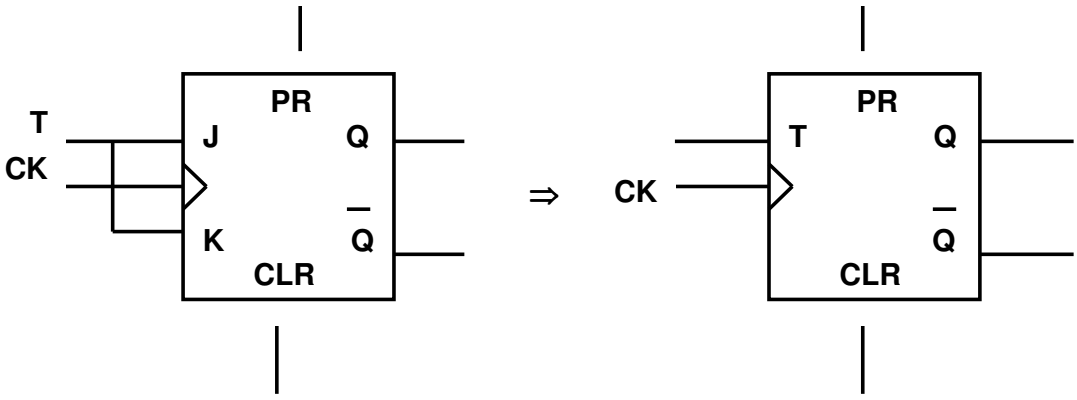
O controle de Preset, quando assumir valor 0, fará com que a saída do circuito (Q) assuma valor 1. O mesmo ocorre com o controle de Clear, fazendo com que a saída assuma valor 0.

A figura mostra o bloco representativo do flip-flop JK Mestre-Escravo com as entradas Preset e Clear ativas em 0.



7.9 Flip-Flop Tipo T

Este flip-flop é obtido de um JK Mestre-escravo com as entradas J e K curto-circuitadas (uma ligada à outra), logo quando J assumir valor 1, K também assumirá valor 1, e quando J assumir valor 0, K também assumirá valor 0. Obviamente, no caso desta ligação, não irão ocorrer nunca entradas como: J = 0 e K = 1; J = 1 e K = 0. A figura mostra a ligação e o bloco representativo do flip-flop tipo T obtido.



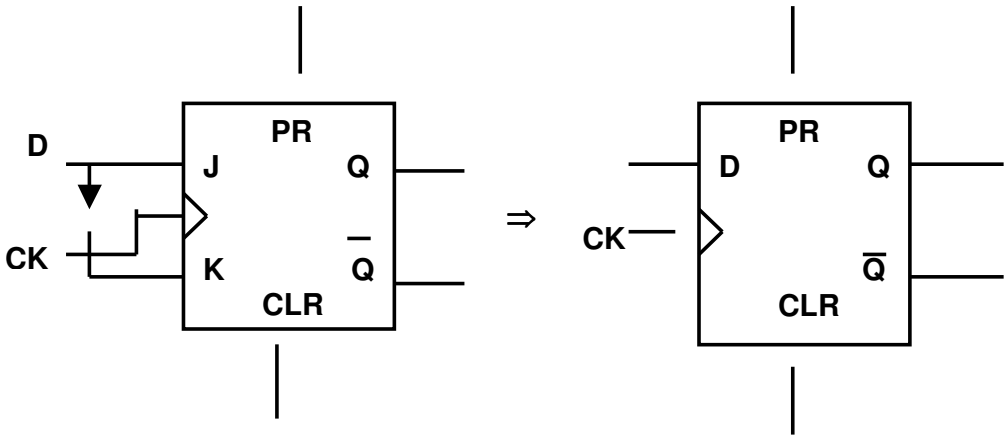
Eliminando os casos não existentes, obtemos a tabela da verdade do flip-flop do tipo T:

T	Qf
0	Qa
1	$\overline{Qa}$

Devido ao fato de o flip-flop tipo T, com a entrada T igual a 1, complementar a saída ($\overline{Qa}$) a cada descida de clock, este será utilizado como célula principal dos **contadores assíncronos** que serão estudados adiante. A sigla **T** vem de **Toggle.** (comutado).

7.10 Flip-Flop Tipo D

É obtido a partir de um flip-flop JK Mestre-Escravo com a entrada K invertida (por inversor) em relação a J. Logo, neste flip-flop, teremos as seguintes entradas possíveis: J = 0 e K = 1; J = 1 e K = 0. Obviamente, não irão ocorrer os casos: J = 0 e K = 0; J = 1 e K = 1. A figura mostra como este é obtido e seu bloco representativo.



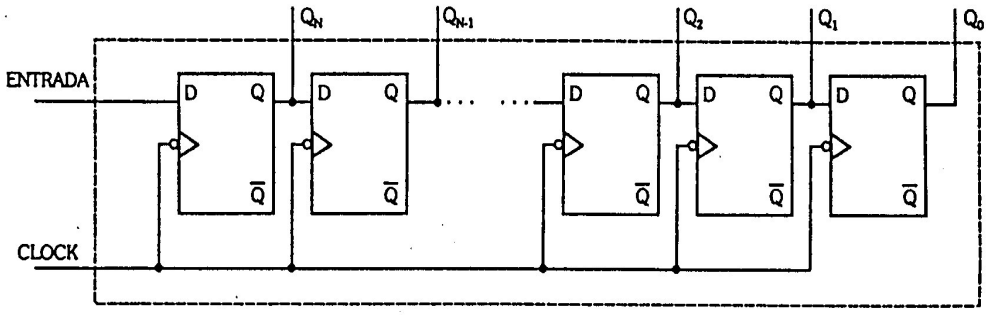
Eliminando os casos não existentes, obtemos a tabela do flip-flop tipo D.

D	Qf
0	0
1	1

Pela capacidade de passar para a saída (Qf) e armazenar o dado aplicado na entrada D, este flip-flop será empregado como célula de **registradores de deslocamento** e em outros sistemas de memória, a serem estudados adiante. A sigla **D** vem de **Data** (dado), termo original em inglês.

7.11 Registradores de Deslocamento

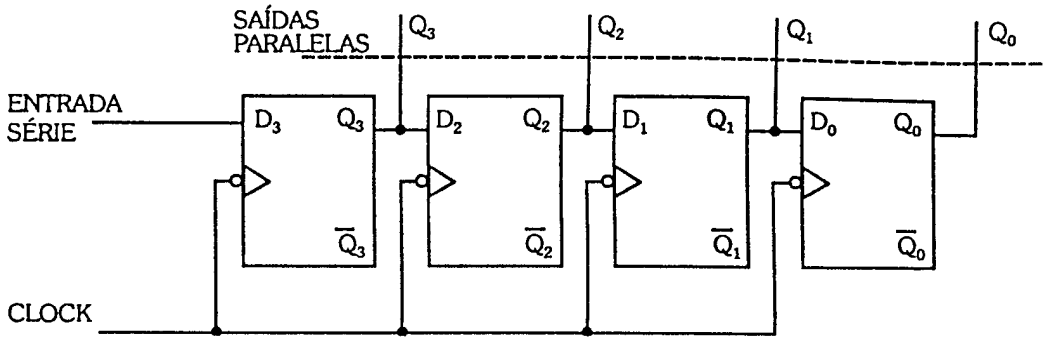
Como vimos, o flip-flop pode armazenar durante o período em que sua entrada clock for igual a 0, um bit apenas (saída Q). Porém, se necessitarmos guardar uma informação de mais de um bit, o flip-flop irá tornar-se insuficiente. Para isso utilizamo-nos de um sistema denominado **Registrador de Deslocamento (Shift Register).** Trata-se de um certo número de flip-flops tipo JK Mestre-Escravo ligado de tal forma que as saídas de cada bloco sejam aplicadas nas entradas J e K respectivas do flip-flop tipo D. A figura representa um registrador de Deslocamento.



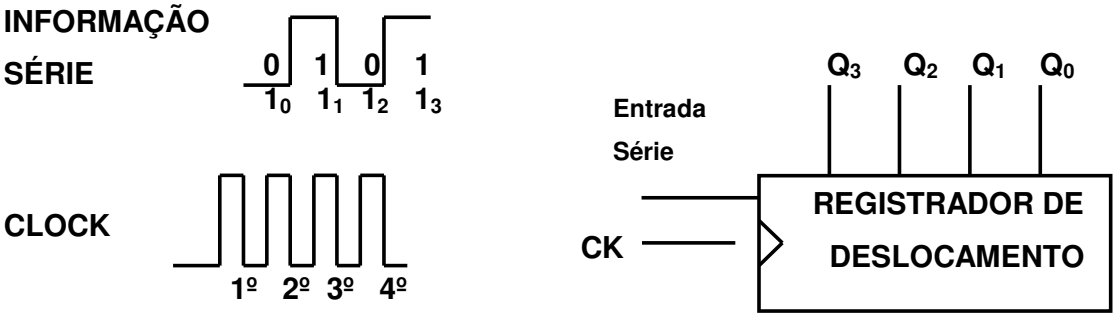
O funcionamento deste sistema, juntamente com suas aplicações, será visto nos itens subseqüente.

7.12 Conversor Série-Paralelo

O Registrador de Deslocamento pode ser usado para converter uma informação série em paralela, ou seja, funcionar como Conversor Série-Paralelo. A configuração básica nessa situação, para uma informação de 4 bits, é vista na figura.



Como exemplo, vamos aplicar a informação série I = 1010 (I_3 I_2 I_1 I_0) à entrada série do registrador e analisar as saídas Q_0, Q_1, Q_2 e Q_3, após os pulsos de clock. Deve-se ressaltar que estes flip-flops atuam como mestre-escravo e têm sua comutação no instante da descida do pulso de clock. Assim sendo, temos:



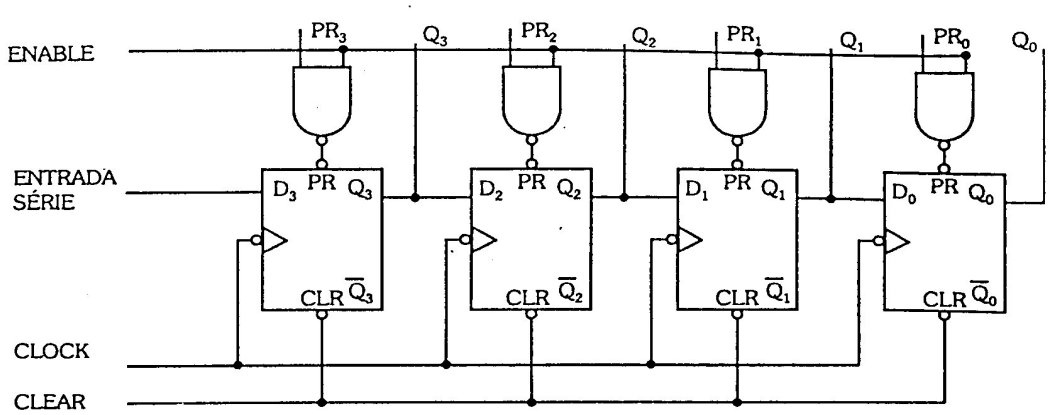
Para resumir, vamos representar toda a seqüência sob a forma da tabela da verdade:

Informação	Descidas de clock	Q_3	Q_2	Q_1	Q_0
I_0 = 0	1ª	0	0	0	0
I_1 = 1	2ª	1	0	0	0
I_2 = 0	3ª	0	1	0	0
I_3 = 1	4ª	1	0	1	0

É pelo motivo de deslocar a informação a cada pulso de clock que esse dispositivo é denominada Registrador de Deslocamento.

7.13 Conversor Paralelo-Série

Para entrarmos com uma informação paralela, necessitamos de um registrador que apresente entradas Prest e Clear, pois é através destas que fazemos com que o Registrador armazene a informação paralela. O registrador com estas entradas é visto na figura.



Primeiramente, vamos estudar o funcionamento da entrada **ENABLE.**

Quando a entrada enable estiver em 0, as entradas preset (PR) dos flip-flops assumirão, respectivamente, níveis 1, fazendo com que o registrador atue normalmente.

Quando a entrada enable for igual a 1, as entradas preset dos flip-flops assumirão os valores complementares das entradas PR_3, PR_2, PR_1 e PR_0, logo, os flip-flops irão assumir os valores que estiverem, respectivamente, em PR_3, PR_2, PR_1 e PR_0.

Para entendermos melhor, vamos analisar uma célula do registrador.

Para zerar (clear) o flip-flop ($Q_3 = 0$), vamos inicialmente, aplicar nível 0 à entrada clear.

Com enable = 0, a entrada PR do flip-flop irá assumir nível 1 e este irá ter um funcionamento normal como célula do registrador de deslocamento em questão, mantendo a saída no estado em que se encontra.

Com enable = 1 e $PR_3 = 0$, a entrada PR do flip-flop assumirá nível 1, logo, a saída Q_3 manterá o seu estado ($Q_3 = 0$). Com enable = 1 e $PR_3 = 1$, a entrada PR do flip-flop assumirá nível 0, forçando a saída a assumir nível 1 ($Q_3 = 1$).

Após essa analise, concluímos que, se zerarmos o registrador (aplicando 0 à entrada clear), e logo após introduzirmos a informação paralela (I_3, I_2, I_1 e I_0) pelas entradas PR_3, PR_2, PR_1 e PR_0, as saídas Q_3, Q_2, Q_1 e Q_0 assumirão respectivamente os valores da informação.

Essa maneira de entrarmos com a informação no registrador é chamada entrada paralela de informação, sendo a entrada responsável pela habilitação da mesma.

Para que o registrador de deslocamento funcione como **Conversor Paralelo-Série,** necessitamos zerá-lo e em seguida, introduzir a informação como já descrito, recolhendo na saída Q_0 a mesma informação de modo série.

É fácil de notar que a saída Q_0 assume primeiramente o valor I_0 e a cada descida do pulso de clock, irá assumir seqüencialmente os valores I_1, I_2 e I_3.

7.14 Contadores

Contadores são circuitos digitais que variam os seus estados, sob o comando de um clock, de acordo com uma seqüência predeterminada. São utilizadas principalmente para contagens diversas, divisões de freqüência, medição de freqüência e tempo, geração de formas de onda e conversão de analógico para digital.

Basicamente, estes sistemas, são divididos em duas categorias: **Contadores Assíncronos e Síncronos.**

➢ **Contadores Assíncronos**

São caracterizados por seus flip-flops funcionarem de maneira assíncrona (sem sincronismo), não tendo entradas clock em comum. Neste tipo de circuito, a entrada clock se faz apenas no primeiro flip-flop, sendo as outras derivadas das saídas dos blocos anteriores.

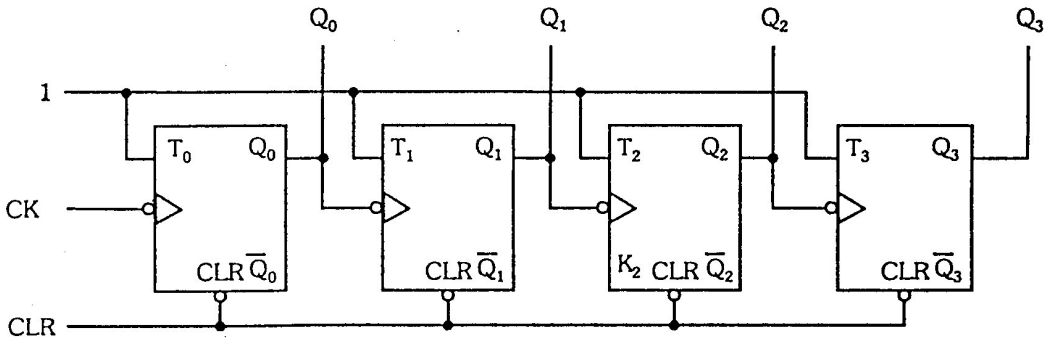
Vamos, a seguir, analisar os principais contadores assíncronos:

● **Contador de Pulsos**

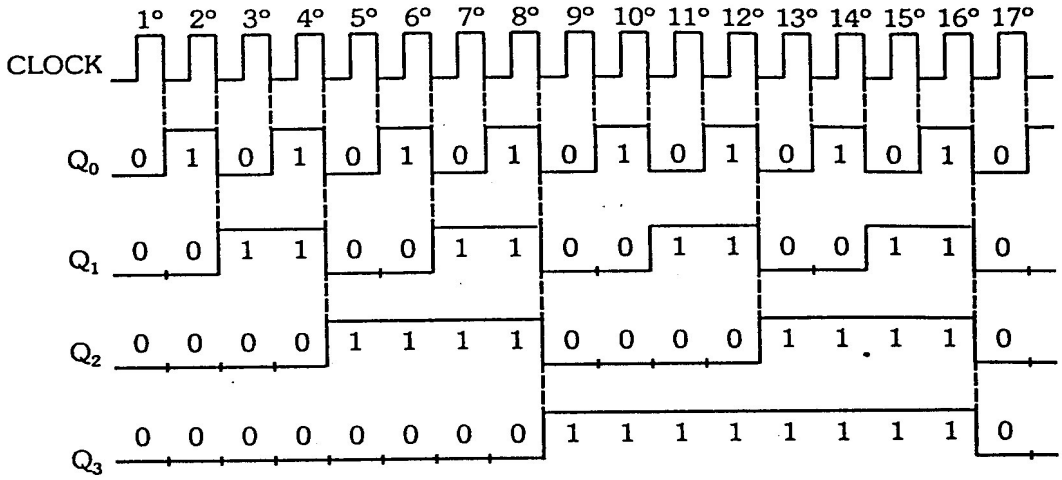
A principal característica de um contador de pulsos é apresentar nas saídas, o sistema binário em seqüência.

Seu circuito básico apresenta um grupo de 4 flip-flops do tipo T ou JK Mestre-Escravo, os quais possuem a entrada T ou, no caso, J e K iguais a 1, originando na saída Qf = $\overline{Q}$a, a cada descida de clock.

A entrada dos pulsos se faz através da entrada clock do 1º flip-flop, sendo as entradas clock dos flip-flops seguintes, conectadas às saídas Q dos respectivos entecessores conforme circuito visto na figura.



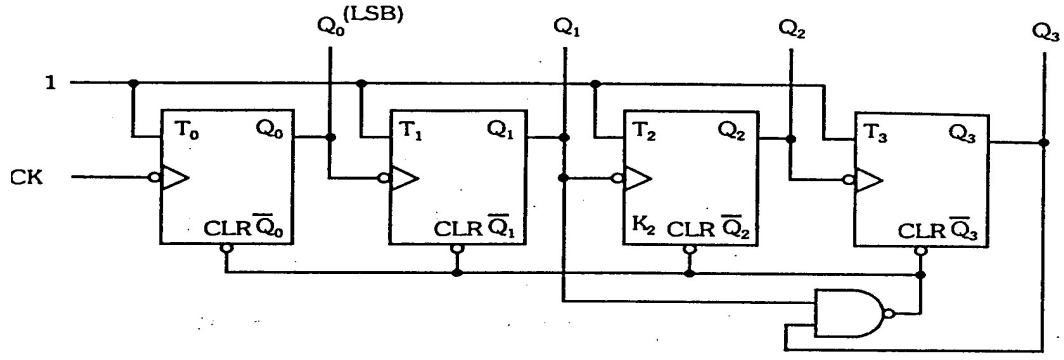
Considerando Q_0 como bit menos significativo (LSB) e Q_3 Como mais significativo (MSB), temos nas saídas o sistema binário em seqüência (0000 a 1111). Notamos ainda, que após a 16ª descida de clock, o contador irá reiniciar a contagem. A figura apresenta toda a seqüência obtida graficamente, a partir da variação aplicada à entrada clock do sistema.



- **Contador de Década**

O contador de década é o circuito que efetua a contagem em números binários de 0 a 9_{10} (10 algarismos). Isso significa acompanhar a seqüência do código BCD 8421 de 0000 até 1001.

Para que o contador conte somente de 0 a 9, deve-se jogar um nível 0 na entrada clear assim que surgir o caso 10 (1010), ou seja, no 10º pulso. O circuito de um contador de década assíncrono é visto na figura.



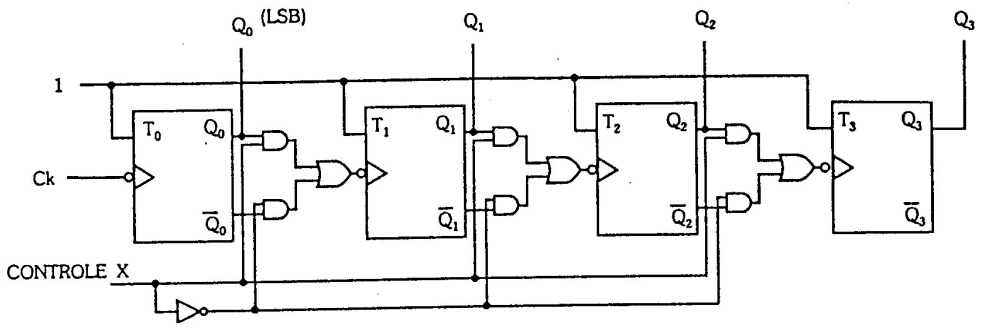
Temos, neste caso, a seguinte tabela da verdade:

Descidas de clock	Q_3	Q_2	Q_1	Q_0	CLR
1ª	0	0	0	0	1
2ª	0	0	0	1	1
3ª	0	0	1	0	1
4ª	0	0	1	1	1
5ª	0	1	0	0	1
6ª	0	1	0	1	1
7ª	0	1	1	0	1
8ª	0	1	1	1	1
9ª	1	0	0	0	1
10ª	1	0	0	1	1
	1	0	1	0	0

Este contador poderá ser utilizado como divisor de freqüência por 10 para uma onda quadrada aplicada à entrada clock, pois possui 10 estados de saída.

- **Contador Assíncrono Crescente/Decrescente**

Podemos construir um contador que execute a contagem crescente ou decrescente. Para isso, utilizamos uma variável de controle que quando assume 1, faz o circuito executar contagem crescente e quando assume 0, faz a contagem decrescente. Este circuito é mostrado na figura.



Notamos que, no circuito, quando o controle X estiver em 1, às saídas $\overline{Q}_0$, $\overline{Q}_1$ e $\overline{Q}_2$ estarão bloqueadas, fazendo com que entrem as saídas Q_0, Q_1 e Q_2 nas entradas clock dos flip-flops respectivamente. Isto fará com que o contador conte crescentemente.

Quando o controle X estiver em 0, a situação investir-se-á e, por conseguinte, o contador contará decrescentemente.

Notamos, ainda, que Q_0 será a saída bit menos significativo (LSB).

O contador crescente/decrescente é também denominado **Up/Down Counter,** que é o termo designativo em inglês.

- **Contadores Síncronos**

Estes contadores possuem entradas clock curto-circuitadas, ou seja, o clock entra em todos os flip-flops simultaneamente, fazendo todos atuarem de forma sincronizada.

Para que haja mudanças de estado, devemos então estudar o comportamento das entradas J e K dos vários flip-flops, para que tenhamos nas saídas, as seqüências desejadas.

Para estudarmos os contadores síncronos devemos sempre escrever a tabela da verdade, estudando quais devem ser as entradas J e K dos vários flip-flops, para que estes assumam o estado seguinte. Para isso, vamos utilizar a tabela da verdade do flip-flop JK.

J	K	Qf	
0	0	Qa	(mantém o estado)
0	1	0	(fixa 0)
1	0	1	(fixa 1)
1	1	$\overline{Q}$a	(inverte o estado)

A partir desta tabela, construímos outra relacionando os estados de saída e as entradas J e K:

	Qa	Qf	J	K
1)	0	0	0	X
2)	0	1	1	X
3)	1	0	X	1
4)	1	1	X	0

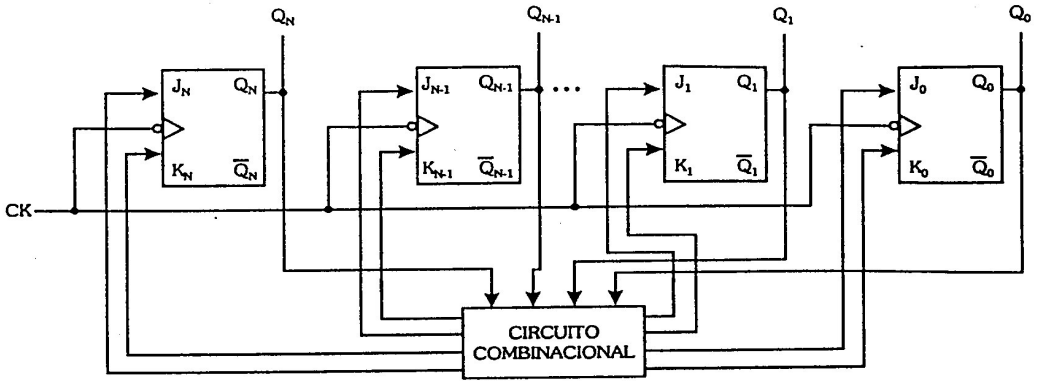
Vamos, a seguir analisar cada caso:

1) Se o flip-flop estiver em 0 (Qa = 0) e quisermos que o estado a ser assumido seja 0 (Qf = 0), podemos tanto manter o estado do flip-flop (J = 0, K = 0 ⇨ Qf = Qa), como fixar 0 (J = 0, K = 1 ⇨ Qf = 0), logo, se J = 0 e K = X, teremos a passagem de Qa = 0 para Qf = 0.
2) Se o flip-flop estiver em 0 (Qa = 0) e quisermos que o estado a ser assumido seja 1 (Qf = 1), podemos tanto inverter o estado (J = 1, K =1 ⇨ Qf = $\overline{Q}$a), como fixarmos 1 (J = 1, K = 0 ⇨ Qf = 1), logo, se J = 1 e K = X, teremos a passagem de Qa = 0 para Qf = 1.
3) Quando o flip-flop estiver em 1 (Qa = 1) e quisermos que ele vá para 0 (Qf = 0), podemos inverter o estado (J = 1, K = 1 ⇨ Qf = $\overline{Q}$a) ou fixar 0 (J = 0, K = 1 ⇨ Qf = 0), logo, se J = X e K = 1, teremos a passagem de Qa = 1 para Qf = 0.

4) Quando o flip-flop estiver em 1 (Qa = 1) e quisermos que ele permaneça em 1 (Qf = 1), podemos manter o estado (J = 0, K = o ⇨ Qf = Qa) ou fixarmos 1 (J = 1, K = 0 ⇨ Qf = 1), logo, se J = X e K = 0, teremos a passagem de Qa = 1 para Qf = 1.

De posse dos resultados das entradas J e K dos flip-flops para a seqüência desejada, obtida da tabela, efetuamos as simplificações e montamos um circuito combinacional que em função das saídas dos flip-flops irá atuar nestas entradas para processar as mudanças de estado.

Genericamente, um contador síncrono possui o esquema visto na figura.

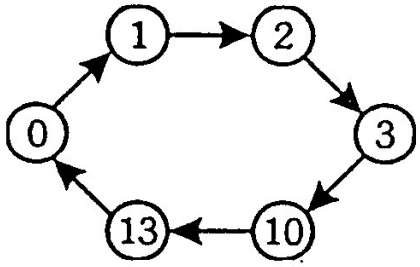


- **Contador Gerador de uma Seqüência Qualquer**

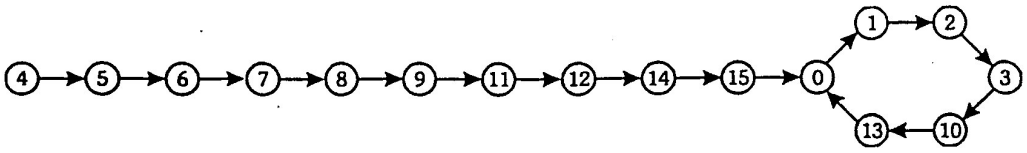
Podemos construir um contador que gere uma seqüência qualquer. Para isso, basta estabelecermos a seqüência e seguirmos o método já conhecido, ou seja, o da determinação das entradas J e K. os estados que não fizerem parte da seqüência deverão ser considerados como condições irrelevantes, ou ser encadeados objetivando atingir o estado inicial.

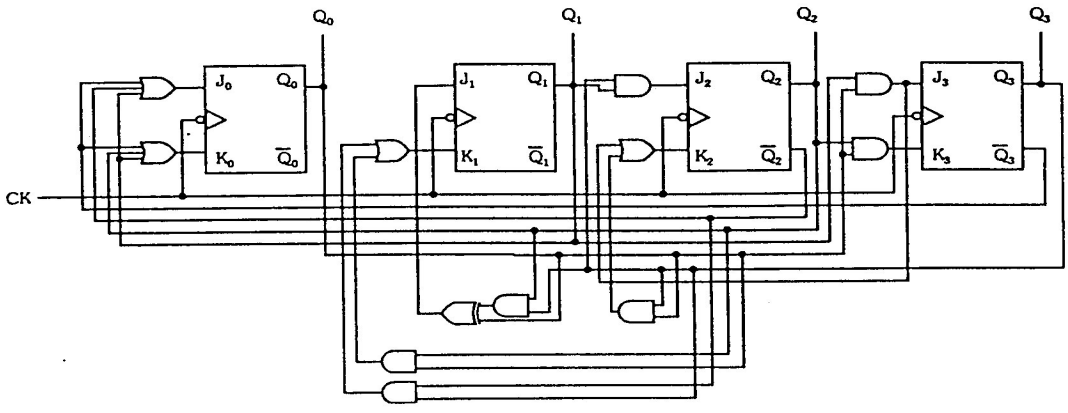
Para exemplificarmos, vamos construir um contador que gere a seguinte seqüência: 0 ⇨ 1 ⇨ 2⇨ 3 ⇨ 10 ⇨ 13 ⇨ 0.

O **loop** que o contador deve efetuar para acompanhar a seqüência é visto no **diagrama de estados** visto na figura.



Notamos que os estados que não pertencem à seqüência são: 4, 5, 6, 7, 8, 9, 11, 12, 14 e 15. Vamos fazer, então, com que o contador, estando no estado 4, após o pulso de clock, vá para o estado 5, deste para o 6 e assim sucessivamente, até que o estado 15 vá para 0 que inicia a seqüência. Esquematicamente, temos:



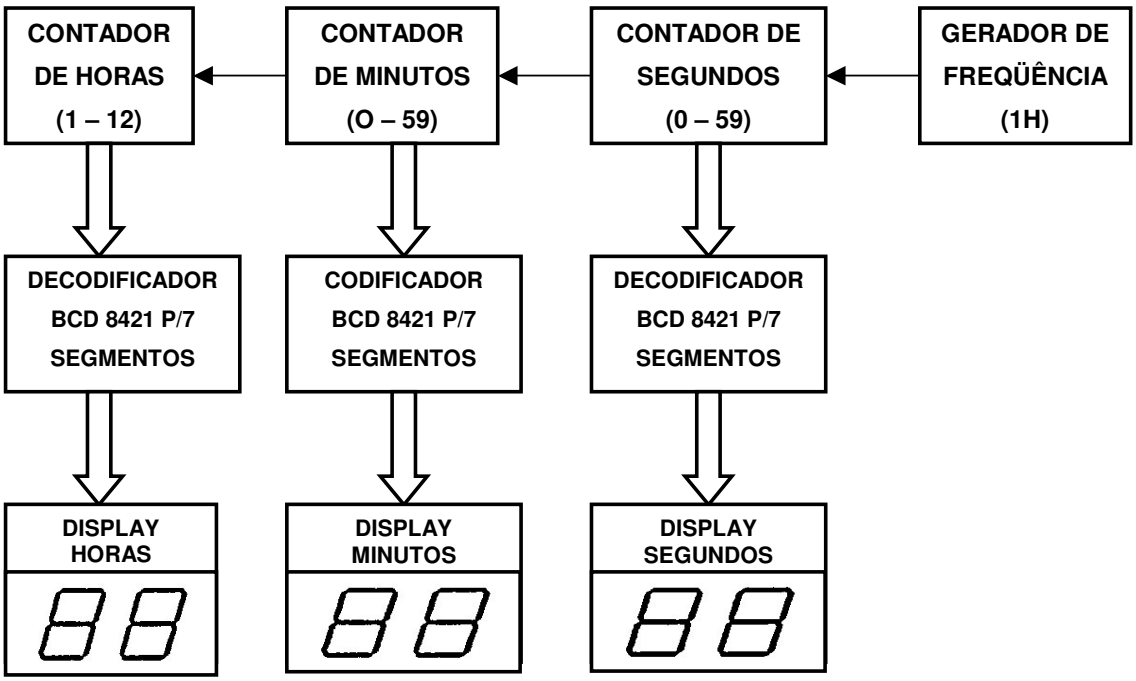


7.15 Contadores Utilizados em Circuitos Temporizadores

Os contadores podem ser usados em várias aplicações nos sistemas digitais. Nos itens subseqüentes, vamos destacar as aplicações em sistemas temporizadores a relógios digitais.

- **Diagrama de Blocos de um Relógio Digital**

Com os elementos vistos até aqui, podemos esquematizar o diagrama de blocos de um relógio digital básico. Este é visto na figura.



8 CIRCUITOS MULTIPLEX E DEMULTIPLEX

8.1 Introdução

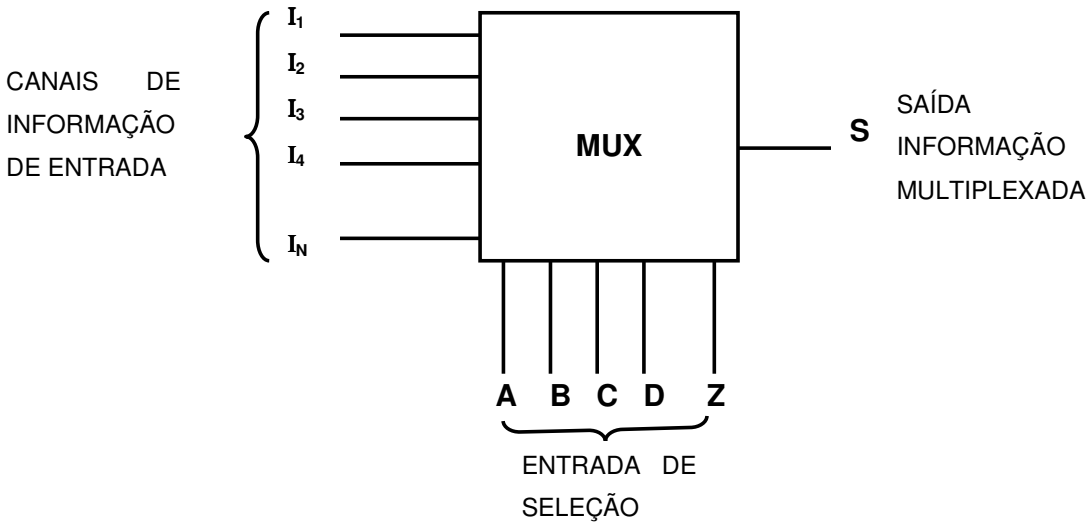
Os circuitos multiplex são utilizados nos casos em que necessitamos enviar em certo número de informações, contidas em vários canais, a um só canal.

Os circuitos demultiplex efetuam a função inversa à dos multiplex, ou seja, enviam as informações, vindas de um único canal, a vários canais.

Ambos os circuitos são largamente empregados dentro de sistemas digitais, bem como na área de Transmissão de dados.

8.2 Multiplex

Como dissemos no início deste capítulo, o circuito multiplex é utilizado para enviarmos as informações contidas em vários canais (fios), a um só canal (fio). Esquematizando o bloco multiplex, temos:



8.3 Projeto do Circuito de um Multiplex

Para projetarmos um multiplex, devemos relacionar, principalmente, a possibilidade de que as entradas de seleção irão assumir com a informação de entrada que deve ser conectada à saída. Para isso, montamos uma tabela da verdade onde serão colocadas todas as possibilidades de seleção e as respectivas informações que devem aparecer na saída.

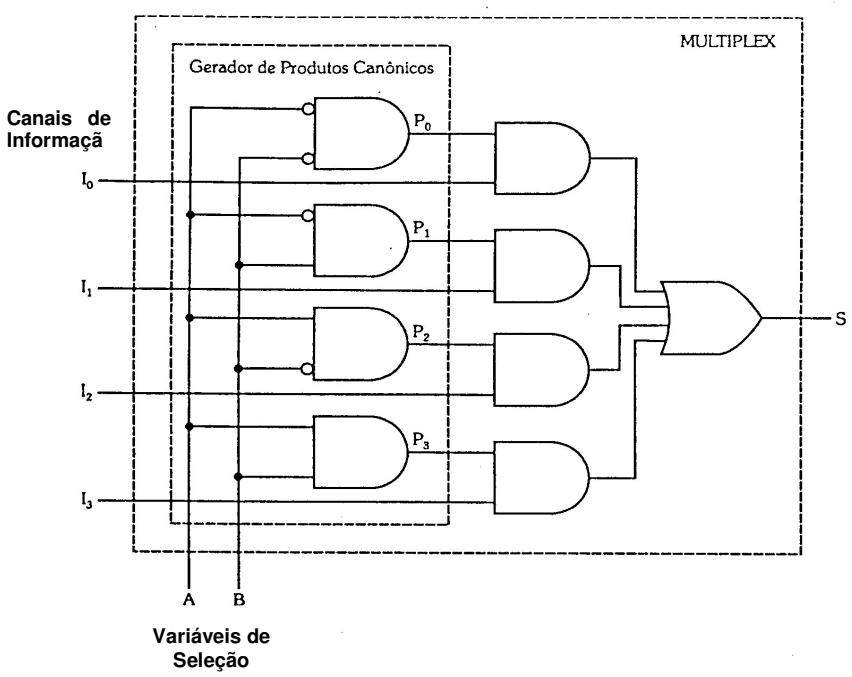
Para mostrarmos passo a passo a elaboração de multiplex, vamos iniciar, efetuando o projeto de um multiplex de 4 canais ou entradas de informações.

Para que possamos conectar aleatoriamente 4 entradas à saída, necessitamos de 2 variáveis de seleção. Com isso, podemos montar a tabela da verdade:

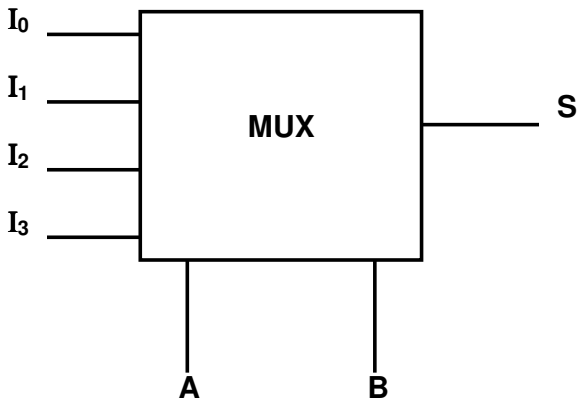
Variáveis de seleção		**Saída**
A	**B**	**S**
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Montando a tabela, relacionamos os valores assumidos pela saída para cada possibilidade das variáveis de seleção, obtendo, a partir disso, o respectivo produto canônico.

Variáveis de Seleção:	**Situação na Saída:**
Caso 00 ($P_0 = \overline{A} \cdot \overline{B}$)	$S = I_0$
Caso 01 ($P_1 = \overline{A} \cdot B$)	$S = I_1$
Caso 10 ($P_2 = A \cdot \overline{B}$)	$S = I_2$
Caso 11 ($P_3 = A \cdot B$)	$S = I_3$

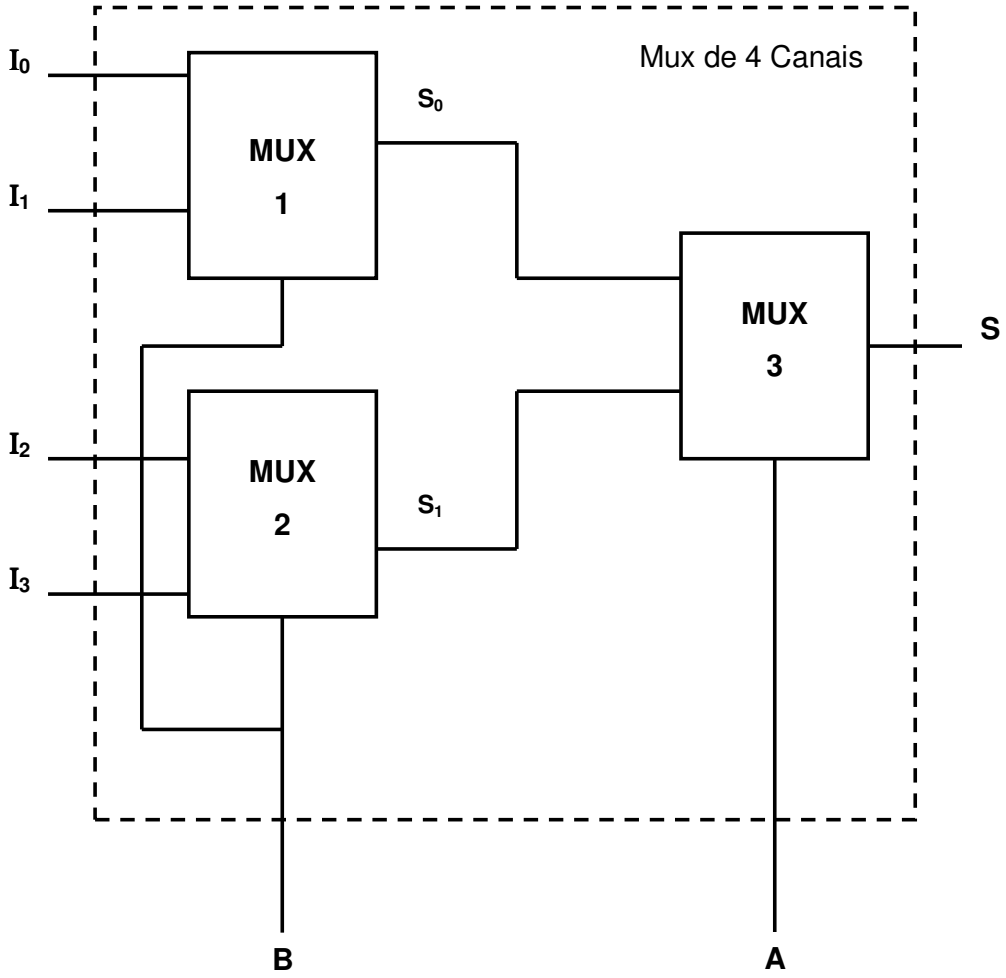


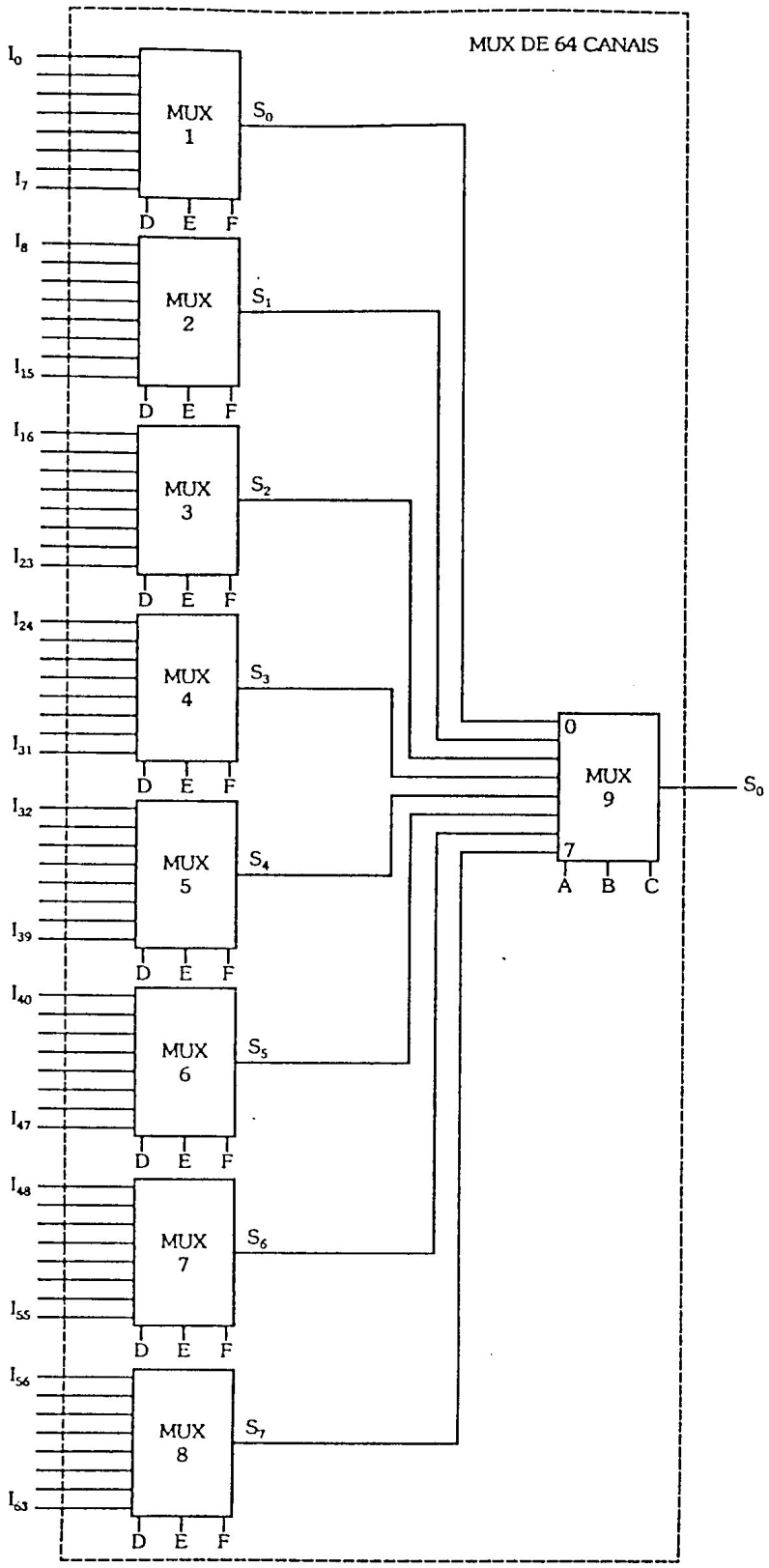
Representando o multiplex obtido em bloco, temos:



8.4 Ampliação da capacidade de um Sistema Multiplex

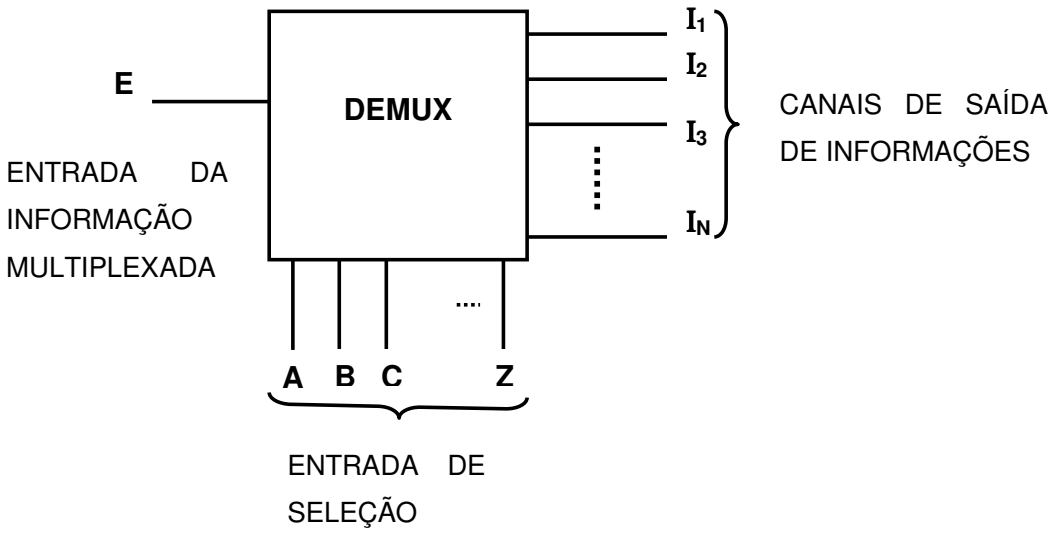
Podemos, a partir de circuitos multiplex de baixa capacidade, formar outros para um maior número de informações de entrada. Para entendermos o processo, vamos montar um multiplex de 4 canais de informação, a partir de outros de apenas 2 canais de informação. A figura mostra, em blocos, o multiplex obtido.


Mux de 4 Canais
I0
I1
I2
I3
MUX
1
MUX
2
MUX
3
S0
S1
S
B
A


MUX DE 64 CANAIS
MUX 1
MUX 2
MUX 3
MUX 4
MUX 5
MUX 6
MUX 7
MUX 8
MUX 9
D E F
A B C
0
7
I_0
I_7
I_8
I_{15}
I_{16}
I_{23}
I_{24}
I_{31}
I_{32}
I_{39}
I_{40}
I_{47}
I_{48}
I_{55}
I_{56}
I_{63}
S_0
S_1
S_2
S_3
S_4
S_5
S_6
S_7
S_0

8.5 Demultiplex

Entende-se por demultiplex como sendo o bloco que efetua a função inversa ao multiplex, ou seja, a de enviar informações contidas em um canal a vários canais de saída. A figura mostra um bloco demultiplex genérico.



As entradas de seleção têm como finalidade escolher qual o canal de informação de saída que deve ser conectado à entrada, ou seja, devem endereçar o canal de saída, ao qual a informação deve se dirigir.

8.6 Projeto do Circuito de um Demultiplex

Para projetarmos um demultiplex devemos relacionar, primeiramente, a possibilidade que as variáveis de seleção irão assumir (endereço), com o canal de saída de informação que deve ser conectado à entrada. Para isso, montamos uma tabela da verdade onde são considerados todas as possibilidades de seleção e os respectivos canais de informação.

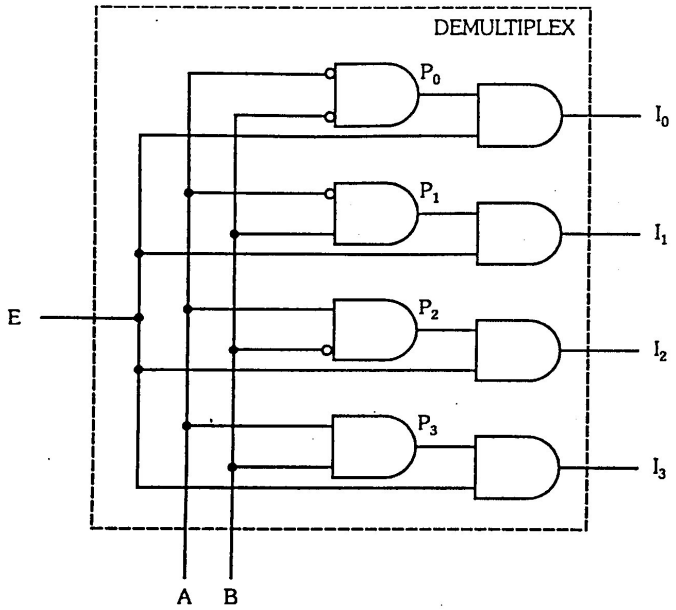
Como exemplo, vamos elaborar um demultiplex de 4 canais. Para que possamos conectar aleatoriamente uma entrada a 4 canais de saída,necessitamos, como já visto, de 2 variáveis de seleção. Com isso, podemos montar a tabela de verdade:

Variáveis		**Canais de Saída**			
A	**B**	I_0	I_1	I_2	I_3
0	0	E	0	0	0
0	1	0	E	0	0
1	0	0	0	E	0
1	1	0	0	0	E

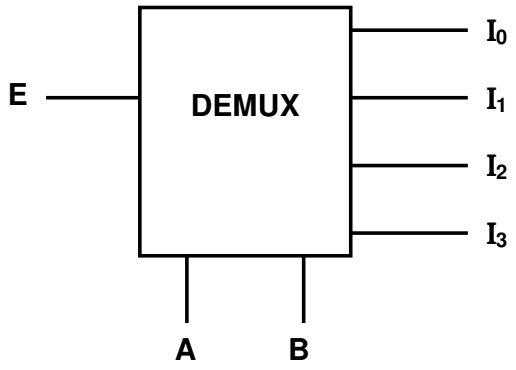
Através de uma tabela, notamos que, quando as variáveis de seleção assumirem:

- ⇨ 00 ($P_0 = \overline{A} . \overline{B}$) : teremos o valor de E no canal de saída I_0.
- ⇨ 01 ($P_1 = \overline{A} . B$) : teremos o valor de E no canal de saída I_1.
- ⇨ 10 ($P_2 = A . \overline{B}$) : teremos o valor de E no canal de saída I_2.
- ⇨ 11 ($P_3 = A . B$) : teremos o valor de E no canal de saída I_3.

O circuito para executar esta função é visto na figura.



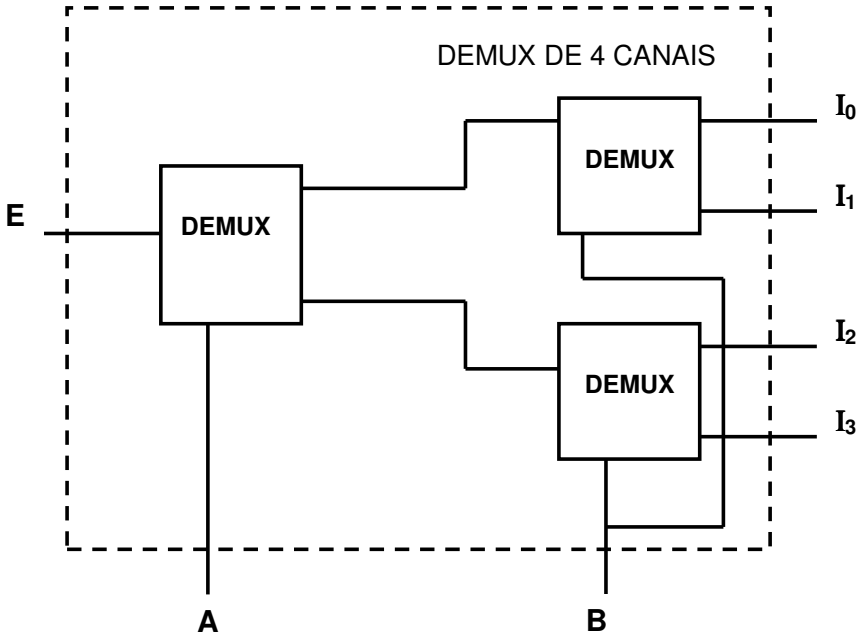
Em bloco, o circuito fica representado:



8.7 Ampliação da Capacidade de um Circuito Demultiplex

Como nos circuitos multiplex, podemos montar a partir de demultiplexadores de menor capacidade, outros de maior capacidade, ou seja, maior número de canais de saída.

Para entendermos o processo, vamos iniciar com um caso simples, onde vamos montar um demultiplex de 4 canais a partir de outros de apenas 2 canais de saída. A figura apresenta esta montagem.



8.8 Multiplex e Demultiplex Utilizados na Transmissão de Dados

Os circuitos Multiplex e Demultiplex são muito utilizados em transmissão de dados. Para isso, basta que tenhamos um bloco no transmissor e um outro no receptor executando a função inversa.

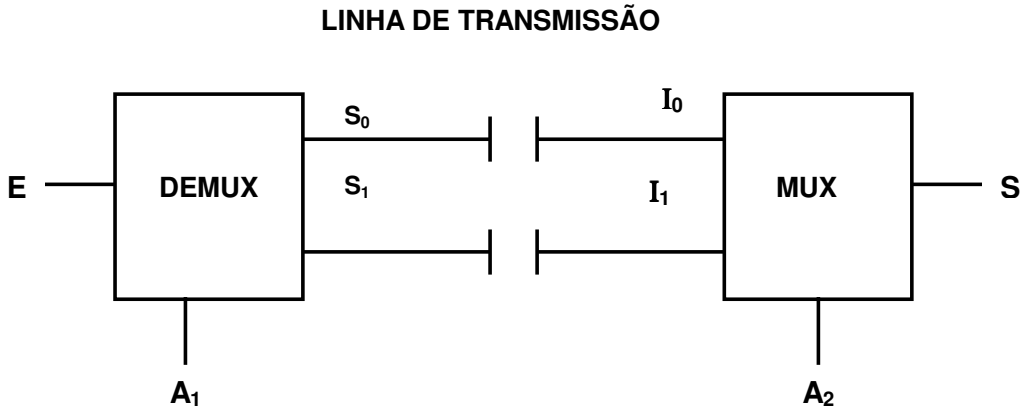
Para que haja uma perfeita recepção, é necessário também que as variáveis de seleção estejam sincronizadas, ou seja, tanto na transmissão como na recepção, as variáveis de controle devem enviar o mesmo endereço. Basicamente, temos dois processos de transmissão:

1 – Transmissão paralela: através de múltiplos fios.
2 – Transmissão série: através de 1 fio.

Vamos, para analisar os processos, exemplificar a transmissão de dados de 2 bits nos dois modos:

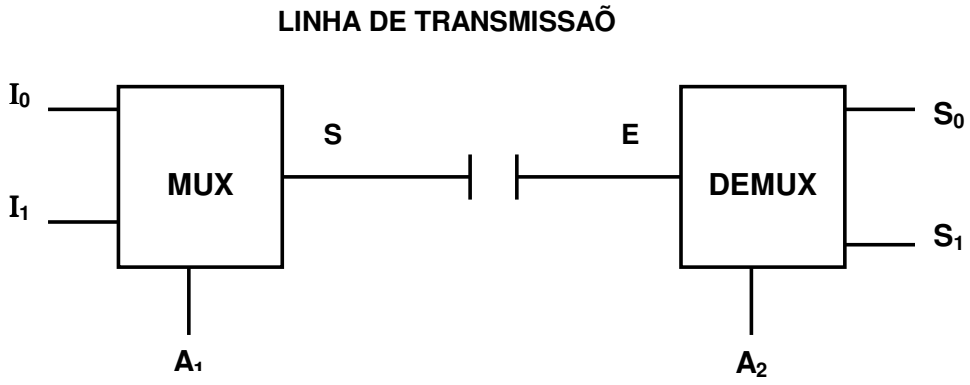
1- Transmissão Paralela

A configuração do circuito neste tipo de transmissão é vista na figura.



2 – Transmissão Série

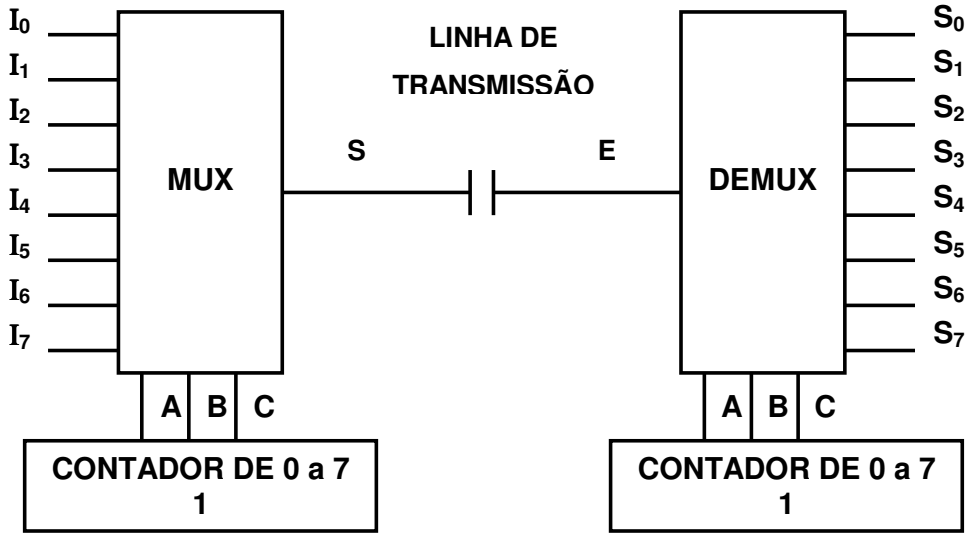
A configuração do circuito é vista na figura.



Neste caso, a entrada da informação é feita por 2 fios (2 bits de informação) e é transmitida através de um único fio. Na recepção, teremos a conversão para saída em 2 fios, como na entrada.

O processo apresenta a vantagem de transmitir a informação de modo série. Este fato é muito importante quando temos uma grande distância entre o transmissor e o receptor, pois a linha de transmissão poderá ser simplesmente um par de fios, linha telefônica ou, ainda, um sistema mais complexo utilizando **fibras ópticas.**

Vejamos a seguir, um sistema de transmissão de dados, utilizando multiplex e demultiplex de 8 canais de informação, ambos com endereçamento seqüencial:



9 FAMÍLIAS DE CIRCUITOS LÓGICOS

As famílias utilizadas atualmente dentro da área de Eletrônica Digital são a **TTL (Transistor-Transistor Logic)** e a **CMOS (Complementary Metal Oxide Semiconductor),** porém derivam de uma série de famílias lógicas, hoje obsoletas.

9.1 Níveis de Tensão e de Corrente

Existe uma terminologia padrão empregada pelos principais fabricantes de circuitos integrados nos respectivos manuais, para designar estes parâmetros. Vamos apresentá-los e defini-los, a seguir:

- ⇨ **V_{IL} (Low-level Input Voltage):** valor de tensão (máxima), que garante o nível 0 na entrada.
- ⇨ **V_{OL}-(Low-level Output Voltage):** valor de tensão (máxima), que garante o nível 0 na saída.
- ⇨ **V_{IH} (High-level Input Voltage):** valor de tensão (Mínima), que garante o nível 1 na entrada.
- ⇨ **V_{OH} (High-level Output Voltage):** valor de tensão (mínima), que garante o nível 1 na saída.
- ⇨ **I_{IL} (Low-level Input Current):** valor de corrente (máxima), no terminal de entrada (no sentido do bloco para o terminal), quando é aplicado o nível 0.
- ⇨ **I_{OL} (Low-level Output Current):** valor de corrente (máxima), que a saída pode receber quando em nível 0.
- ⇨ **I_{IH} (High-level Input Current):** valor de corrente de entrada (máxima), quando é aplicado nível 1.
- ⇨ **I_{OH} (HIGH-LEVEL Output Current):** valor de corrente de saída (máxima), quando em nível 1.
- ⇨ **I_{OH} (High-level Output Current):** valor de corrente de saída (máxima), quando em nível 1.

Nos manuais, além dos limites de mínimo e máximo, conforme a definição do parâmetro, são encontrados os valores típicos de trabalho.

9.2 Características Gerais e parâmetros da Família TTL

Os circuitos TTL são produzidos em duas séries comerciais: a série 74XXX e 54XXX, sendo esta última denominada série militar ou profissional, devido à maior margem de variação nas especificações de alimentação e temperatura, assegurando a confiabilidade no desempenho em condições máximas.

Vamos agora, enumerar os principais parâmetros encontrados nos manuais em nomenclaturas originais:

1- Alimentação (V_{CC}): Na família TTL, temos para todos os blocos uma alimentação de 5V. Para a série 54 temos V_{CC} mínimo = 4,5V e V_{CC} máximo = 5,5V que são valores dentro da especificação militar de 10% de tolerância. Para a série de 74, temos V_{CC} mínimo = 4,75V e V_{CC} máximo = 5,25V que são valores dentro da especificação comum de 5% de tolerância.

2- Níveis de entrada e saída, para a versão padrão (**TTL Standard):**

TTL	**Standard**	
Parâmetros	**Valores**	**Unidade**
V_{IL}	0,8	V
V_{OL}	0,4	V
V_{IH}	2,0	V
V_{OH}	2,4	V
I_{OL}	16	mA
I_{IL}	1,6	mA
I_{OH}	400	µA
I_{IH}	40	µA

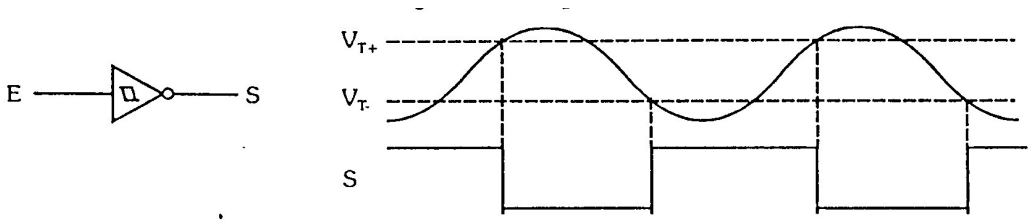
- **Schimitt-Trigger**

São também encontrados disponíveis na família TTL, blocos configurados com entradas Schimitt-Trigger (gatilho de Schimitt).

Este tipo de bloco possibilita tornar rápidas, as variações lentas dos níveis de tensão de determinados sinais aplicados à sua entrada, causando na saída o aparecimento de uma onda quadrada bem-definida.

Em outras palavras, este tipo de bloco, além de realizar sua função lógica, quadra o sinal aplicado à entrada, desde que sejam respeitados os parâmetros mínimos e máximos de tensão especificados para o bloco.

Para ilustrar, a figura apresenta um inversor TTL schimitt-trigger (a) e a ação sobre um sinal de variação lenta aplicado à sua entrada (b).



- **Tri-state**

Conforme já visto, existem blocos que apresentam um terceiro estado de saída (tri-state) caracterizado por uma alta impedância.

9.3 Versões dos Circuitos TTL

Além dos blocos comuns (Standard), a família TTL possui outras versões de circuitos com a finalidade de atender a solicitações de ordem prática nos parâmetros relativos à velocidade e consumo de potência. A tabela apresenta um quadro comparativo entre estas versões e as respectivas identificações.

Versão	Identificação da série	Tempo de atraso de propagação típico por porta	Consumo de potência por porta	Freqüência de clock máxima para flip-flop	Observações
Standard	54/74	10 ns	101 mW	35 MHz	Comum
Low power	54L/74L	33 ns	1 mW	3 MHz	Baixíssimo consumo
High speed	54H/74H	6 ns	22 mW	50 MHz	Alta velocidade
Schottky	54S/74S	3 ns	19 mW	125 MHz	Altíssima velocidade
Advanced Schottky	54LS/74LS	1,5 ns	8,5 mW	200 MHz	Altíssima velocidade e baixo consumo
Low power Schottky	54LS/74LS	10 ns	2 mW	45 MHz	Baixíssimo consumo
Advanced Low power Schottky	54LS/74LS	4 ns	1 mW	70 MHz	Altíssima velocidade e baixíssimo consumo

9.4 Características Gerais e Parâmetros da Família CMOS

A família CMOS possui circuitos integrados disponíveis nas séries comerciais 4000A, 4000B e 54/74C, sendo esta última semelhante a TTL na pinagem dos circuitos integrados e função dos blocos disponíveis.

Além destas, a família CMOS também possui versões de alta velocidade e melhor desempenho: 74HC/74HCT **(High-speed CMOS)**, sendo a HCT especialmente desenvolvida para atuar com parâmetros de tensões compatíveis com TTL-LS, e as apropriadas para operar com baixa tensão de alimentação: 74LV/74LVC **(Low Voltage CMOS).**

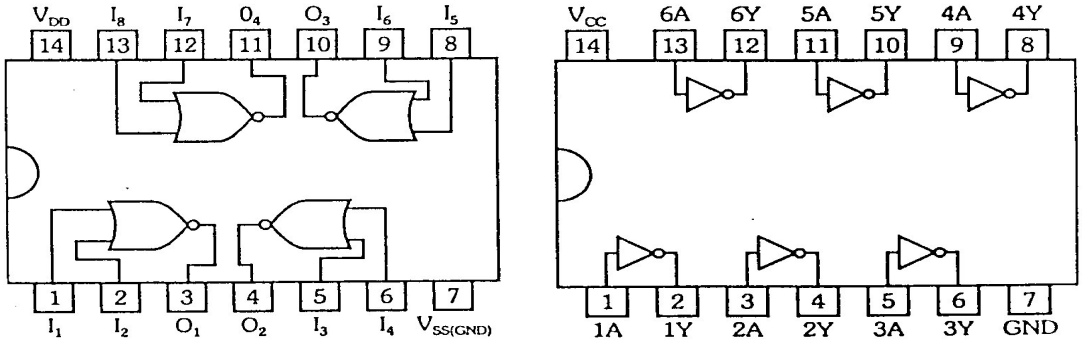
A tabela apresenta os valores de tensões e correntes para a série 4000B, operando com V_{DD} igual a 5V.

CMOS 4000B		
Parâmetros	**Valores**	**Unidade**
V_{IL}	1,5	V
V_{OL}	0,05	V
V_{III}	3,5	V
V_{OII}	4,95	V
I_{OL}	0,4	mA
I_{IL}	1	µA
I_{OII}	0,4	mA
I_{III}	1	µA

9.5 Circuitos Integrados CMOS

Da mesma forma que na TTL, a família CMOS colocou no mercado uma série de circuitos integrados padronizados com configurações de pinagens disponíveis nos manuais dos fabricantes.

Para exemplificar, a figura apresenta a pinagem do circuito integrado 4001B (4 NOU com 2 entradas) e do 74HC04/74HCT04 (6 inversores), sendo estes últimos de mesma pinagem que o 7404 da família TTL.



(a) 4001B

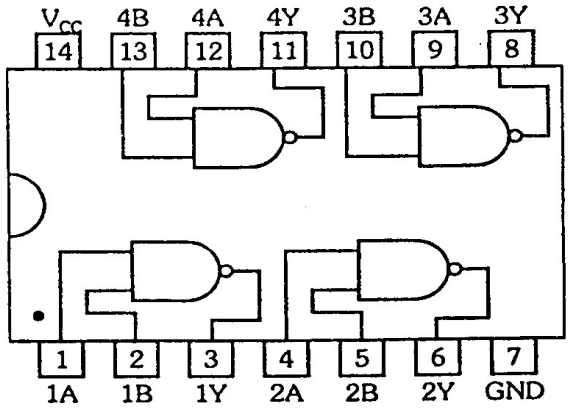
(b) 74HC04/74HCT04

9.6 Circuitos Integrados TTL

A família TTL colocou no mercado uma série de circuitos integrados padronizados com configurações de pinagens disponíveis nos manuais dos fabricantes.

São circuitos integrados de 14 pinos ou mais, conforme a complexidade do circuito agregado, com encapsulamentos denominados **DIP (Dual-In-Line-Package)**, cuja identificação da disposição dos terminais se faz através da vista superior, em sentido anti-horário, a partir do ponto de referência colocado no pino 1, próximo ao chanfro existente no bloco.

Para exemplificar, a figura apresenta a pinagem do circuito integrado 740 (4 NE com 2 entradas), sendo esta válida também para o 5400 e, ainda, para as versões 74L00, 74H00, 74S00, 74AS00, 74LS00,74ALS00.



Alimentação:

Pino 14: $+V_{CC}$

PINO 7: terra ou GND (ground)

10 ENSAIOS DE ELETRÔNICA DIGITAL

10.1 Ensaio 1 (Introdução a Eletrônica Digital)

- **Objetivo**

Montar portas lógicas a partir de componentes discretos;
Construir tabelas verdades das diversas portas.

- **Informação Teórica**

Embora a eletrônica digital moderna seja totalmente construída de componentes integrados (CI), utilizaremos neste primeiro contato, diodos e transistores sob forma discreta.

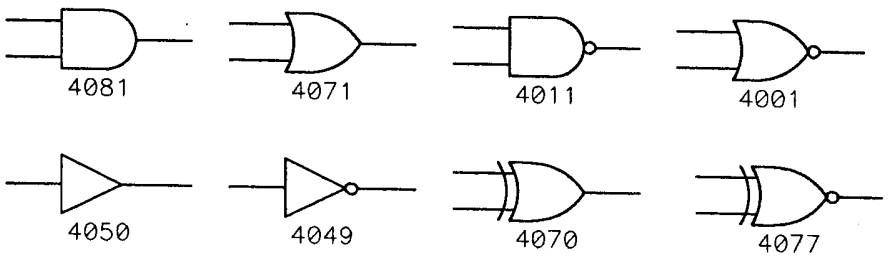
Toda eletrônica digital está montada na combinação de Portas Lógicas básicas, que serão vistas neste ensaio, e unidades de Memória, denominadas de Flip-Flop, que serão estudas posteriormente.

A eletrônica digital, diferentemente da eletrônica linear, caracteriza-se pela utilização de dois estados definidos, aos quais podemos associar:

Não	Bloqueia	Cortado	Low	Baixo	0V	0V	0

Sim	Conduz	Saturado	High	Alto	5V	12V	1

Uma porta lógica consiste num bloco que possui duas ou mais entradas que assumem um dos dois estados, mostrados acima, denominados de **BIT (Binary Digit)**. A porta fornece na saída um nível de acordo com o seu comportamento lógico característico, denominado de *tabela verdade.*

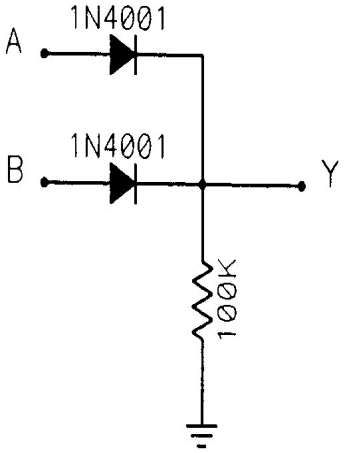


- **Material a ser Utilizado**

1 Transistor BC 548
2 Diodos IN4001
2 Resistores de 15K
1 Resistores de 100

- **Procedimentos**

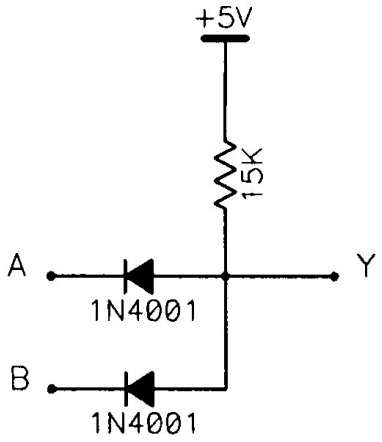
1. Monte o circuito abaixo, observando com atenção a polaridade dos diodos. Ligue a saída Y a um dos indicadores lógicos e as entradas A e B a duas chaves de dados. Observe o indicador lógico e complete a tabela verdade.
Que tipo de Porta Lógica é o circuito abaixo?



A	B	Y
0	0	
0	1	
1	0	
1	1	

2. Monte agora o circuito abaixo e construa a sua tabela verdade.

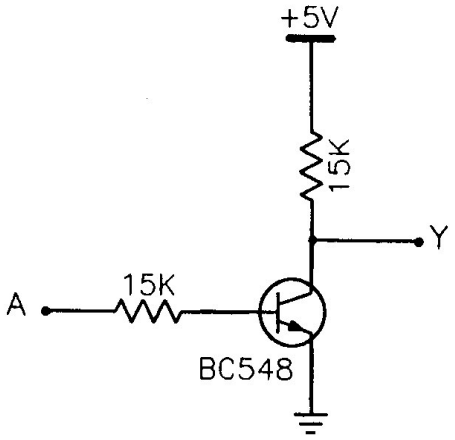
Que tipo de Porta Lógica é o circuito abaixo?



A	B	Y
0	0	
0	1	
1	0	
1	1	

3. Monte o circuito mostrado a seguir e construa a sua tabela verdade.

Que tipo de Porta Lógica é o circuito abaixo?



A	Y
0	
1	

4. Acople diretamente a saída do circuito do item 1 à entrada do circuito do item 3. Desenhe o circuito e construa sua tabela verdade.

Que tipo de Porta Lógica é o circuito abaixo?

A	B	Y
0	0	
0	1	
1	0	
1	1	

5. Acople diretamente a saída do circuito do item 2 à entrada do circuito do item3. Desenhe o circuito e construa sua tabela verdade.
Que tipo de Porta Lógica é o circuito abaixo?

A	B	Y
0	0	
0	1	
1	0	
1	1	

10.2 Ensaio 2 (Portas Lógicas)

- **Objetivo**

1. Usar circuitos com portas E, OU, NE e NOT para determinar as relações entre as entradas e saídas.

2. Construir a tabela verdade de combinações de portas E, OU, NE, NOU e NOT a partir de resultados experimentais.

3. Escrever a expressão boolena para circuitos de lógica combinatória.

4. Analisar outras combinações de portas E, OU, NE, NOU.

5. Identificar pinagem dos Circuitos Integrados.

6. Construir circuitos digitais utilizando circuitos integrados com tecnologia TTL e CMOS.

7. Achar defeitos neste tipo de circuito.

- **Precauções**

1. Deve-se tomar bastante cuidado ao se lidar com Circuitos Integrados visto que seus terminais são muito delicados.

2. Evite, o máximo possível, o contato direto dos dedos com os terminais dos CT's tipo CMOS, devido a possível eletrostáticas, que poderão danificar irreversivelmente o CI.

3. Nunca se deve ligar um CI de maneira errada, ou seja, alimentar a entrada positiva com uma tensão negativa e vice-versa.

Observe sempre a seguinte polarização:

- V_{SS}, V_{EE}, GND $\Rightarrow$ ligar ao – (negativo) da fonte;
- V_{CC}, V_{DD} $\Rightarrow$ ligar ao + (positivo) da fonte.

4. Alimentação dos CI's:

- TTL $\Rightarrow$ 5V $\pm$ 5%
- CMOS $\Rightarrow$ 5V a 15V

- **Material a ser Utilizado**

CI diversos

OBSERVAÇÕES:

1. Se necessário usar combinações de portas, para conseguir a lógica de uma porta especifica;

2. Na montagem de um circuito lógico, implementar com CI's de uma mesma família (CMOS ou TTL).

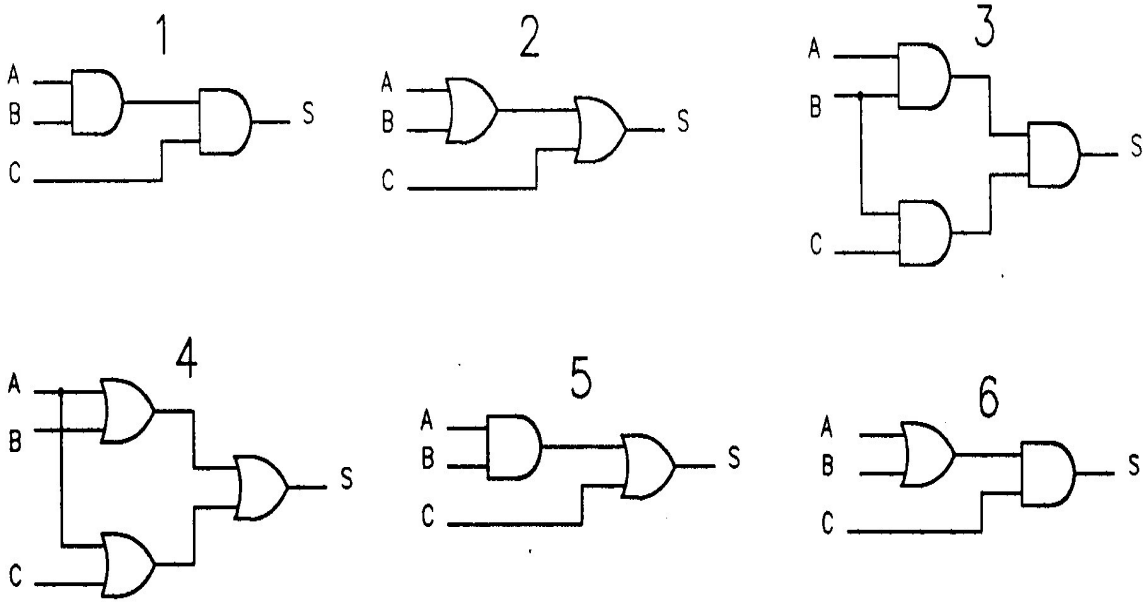
- **Procedimentos**

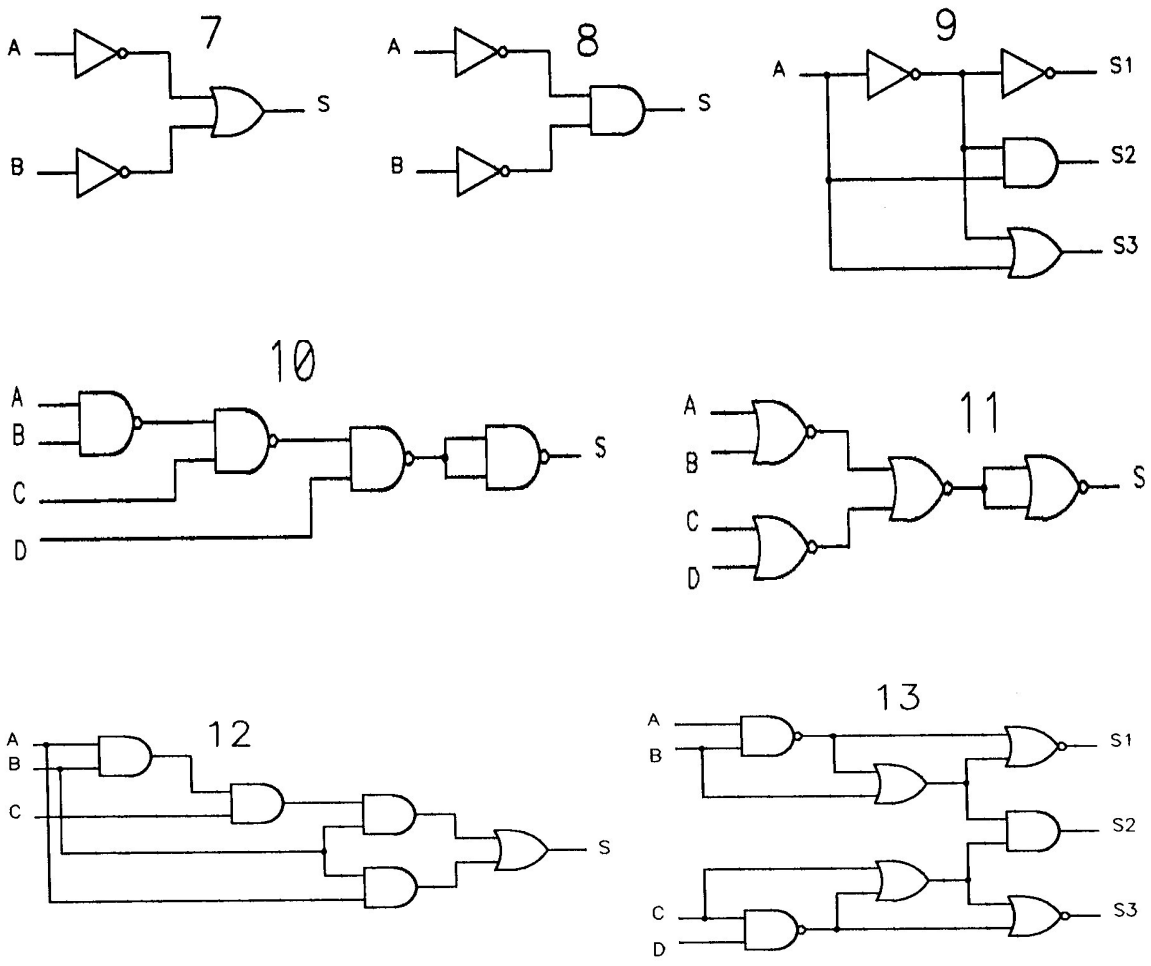
1. Determine a tabela verdade do circuito lógico Nº1.

2. Escreva a expressão booleana do circuito lógico Nº1.

3. Monte o circuito lógico Nº1 e confira o resultado com a tabela da verdade levantada.

4. Repita os procedimentos anteriores para os demais circuitos lógicos.





10.3 Ensaio 3 (Álgebra Booleana)

- **Objetivo**

1. Montar circuitos combinacionais e obter suas tabelas verdade;

2. Comprovar diversos teoremas da Álgebra Booleana.

- **Informação Teórica**

Em meados do século XIX, George Boole, desenvolveu um sistema de analise lógica que hoje é conhecido com Álgebra Boleana. Suas propriedades, postuladas e teoremas permitem a síntese e a otimização de circuitos digitais.

Através do domínio da Álgebra Booleana, podemos obter uma maior visão para solução de determinados circuitos, bem como obteremos margens para improvisações, às vezes bastante lucrativas.

Mas o verdadeiro valor da Álgebra Booleana está na simplificação de circuitos de equações lógicas. Um circuito pode ser otimizado, o que implica numa economia de material e trabalho, de montagem e teste.

- **Material a ser Utilizado**

CI diversos

OBSERVAÇÕES:

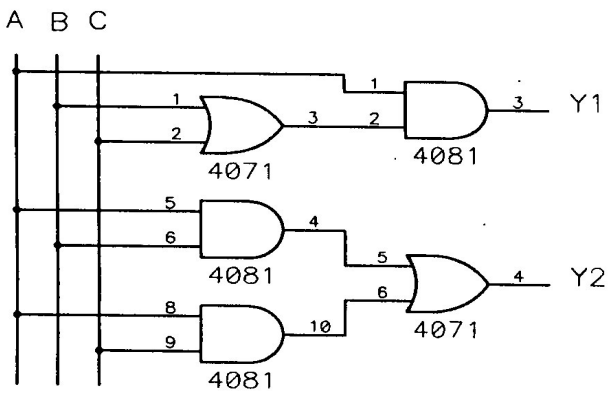
1. Se necessário usar combinações de portas, para conseguir a lógica de uma porta especifica;

2. Na montagem de um circuito lógico, implementar com CI's de uma mesma família (CMOS ou TTL).

- **Procedimentos**

Para cada um dos procedimentos seguintes, montar os dois lados da igualdade e comprovar o Teorema levantando a tabela verdade para os dois circuitos ao mesmo tempo.

1º Teorema: A(B + C) = AB + AC



A	B	C	Y1	Y2
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

2º Teorema: $\overline{A + B} = \overline{A}\ \overline{B}$

A	B	Y1	Y2
0	0		
0	1		
1	0		
1	1		

3º Teorema: $A + AB = A$

A	B	Y1	Y2
0	0		
0	1		
1	0		
1	1		

4º Teorema: $A + \overline{A}B = A + B$

A	B	Y1	Y2
0	0		
0	1		
1	0		
1	1		

5º Teorema: $A + BC = (A + B)(A + C)$

A	B	C	Y1	Y2
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

6º Teorema: $\overline{AB} = \overline{A} + \overline{B}$

A	B	Y1	Y2
0	0		
0	1		
1	0		
1	1		

7º Teorema: A(A + B) = A

A	B	Y1	Y2
0	0		
0	1		
1	0		
1	1		

8º Teorema: A($\overline{A}$ + B) = AB

A	B	Y1	Y2
0	0		
0	1		
1	0		
1	1		

10.4 Ensaio 4 (Circuitos Combinacionais)

- **Objetivo**

Determinar um circuito lógico para executar uma função desejada.

- **Introdução**

O circuito combinacional é aquele em que a saída depende única e exclusivamente das combinações entre as variáveis de entrada.

Podemos utilizar um circuito combinacional para solucionar problemas em que necessitamos de uma resposta, quando acontecerem determinadas situações, representadas pelas variáveis de entrada.

Para construirmos estes circuitos, necessitamos de suas expressões características que são obtidas das tabelas verdades que representam as situações mencionadas.

- **Material a ser Utilizado**

CI's diversos

OBSERVAÇÕES:

1. Se necessário usar combinações de portas, para conseguir a lógica de uma porta especifica.

2. Na montagem de um circuito lógico, implementar com CI's de uma mesma família (CMOS) ou TTL).

- **Procedimentos**

1. Estabeleça qual a função que se deseja realizar, definindo as entradas e saídas envolvidas;

2. Coloque o problema em termos de uma tabela da verdade ou equação lógica que forneçam a saída ou saídas desejadas em função das entradas disponíveis;

3. Obtenha o circuito a partir da tabela verdade ou das equações encontradas no passo anterior, procurando-se simplificar estes circuitos tanto quanto possível;

4. Implemente na pratica o circuito, levando em consideração fatores como consumo de potência, velocidade requerida, componentes disponíveis, etc.

- **Projetos de Circuitos Combinacionais**

1. Circuito com 2 variáveis

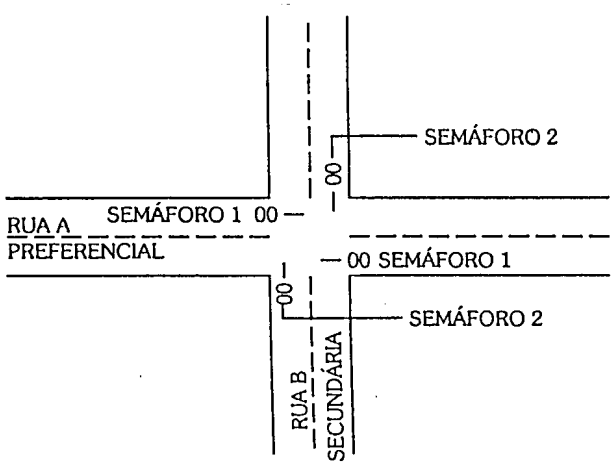
A figura abaixo representa o cruzamento das Ruas “A” e “B”. Neste cruzamento, queremos instalar um sistema automático para os semáforos, com as seguintes características:

1. Quando houver carros transitando somente na rua B, o semáforo 2 deverá permanecer Verde para que os automóveis possam trafegar livremente.

2. Quando houver carros transitando somente na rua A, o semáforo 1 deverá permanecer Verde pelo mesmo motivo.

3. Quando não houver carros transitando nas A e B, devemos abrir o semáforo para rua A.

4. Quando houver carros transitando nas ruas A e B, devemos abrir o semáforo para rua A, pois é preferencial.



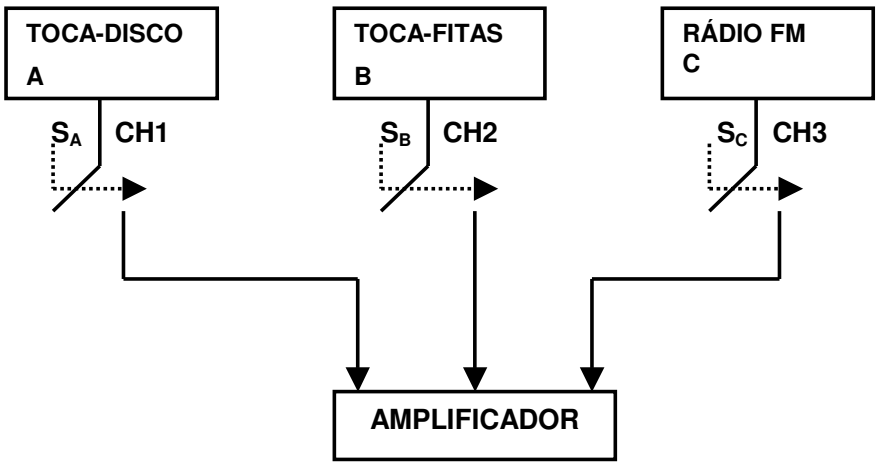
2. Circuitos com 3 variáveis

Deseja-se utilizar um amplificador para ligar três aparelhos: um toca-fitas, um toca-disco e um rádio FM. Vamos elaborar um circuito lógico que nos permita ligar os aparelhos, obedecendo às seguintes prioridades:

1ª prioridade: Toca-disco
2ª prioridade: Toca-fitas
3ª prioridade: Rádio FM

Isto significa que quando não ligarmos nem o toca-disco, nem o toca-fitas, o rádio FM, se ligado, será conectado à entrada do amplificador. Se ligarmos o toca-fitas, automaticamente o circuito irá conecta-lo à entrada do amplificador, pois possui prioridade sobre o rádio FM. Se, então ligarmos o toca-disco, este será conectado ao amplificador, pois representa a 1ª prioridade. A partir disto, podemos montar o diagrama de blocos com as respectivas ligações:

Neste projeto, o circuito lógico receberá as informações das variáveis de entrada A,B e C, representando os aparelhos, e através das saídas S_A, S_B, S_C comutará as chaves CH1, CH2 e CH3 para fazer a conexão conforme a situação requerida.



10.5 Ensaio 5 (Decodificador BCD 8421 para Display de 7 Segmentos)

● Objetivo

Montar um decodificador BCD para 7 segmentos usando um display com LEDs.

● Introdução

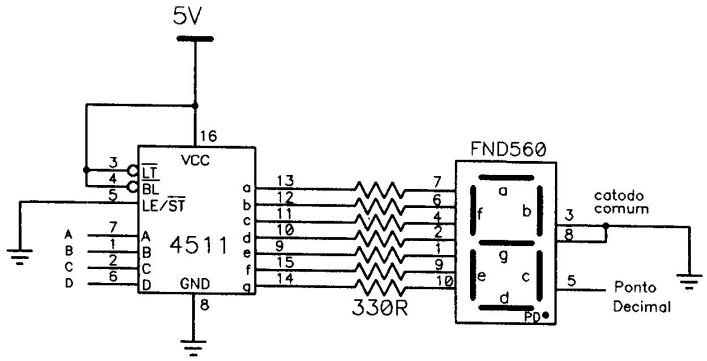
Um decodificador nada mais é do que um tradutor que, em nosso caso traduzirá o código BCD (Banary Coded Decimal), que significa uma codificação do sistema decimal em binário, em 7 segmentos, que formam um display com leds e nos possibilita escrever números de 0 a 9 e alguns outros símbolos que podem ser letras ou sinais.

● Material a ser Utilizado

1 CI 4511
1 Display com catodo comum
7 resistores 330Ω

● Procedimentos

1. Monte o circuito abaixo, observando com muita atenção a perfeita localização de cada pino.



2. Ligue o circuito e verifique se o circuito realmente obedece à tabela verdade de um decodificador BCD para 7 segmentos.

3. Desligue o pino 3 do +5V e ligue-o ao GND. O que acontece?

4. Repita o procedimento para os pinos 4 e 5 individualmente. O que acontece?

10.6 Ensaio 6 (Circuitos Aritméticos)

- **Objetivo**

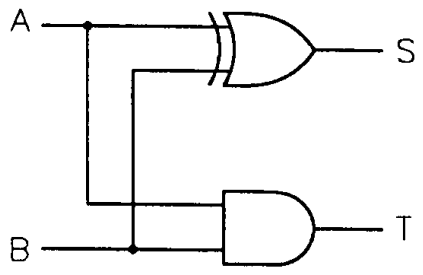
1. Construir um circuito somador completo básico;
2. Implementar um somador de 4bits.

- **Informação Teórica**

Uma das grandes aplicações dos circuitos lógicos está nas unidades aritméticas dos sistemas digitais, onde são realizadas as operações em código binário.

O circuito básico é o somador binário. Sua importância reside no fato de que em muitas unidades aritméticas, pode-se realizar qualquer operação matemática. A subtração, por exemplo, é obtida por um processo de soma de complementos, a multiplicação por soma e deslocamentos sucessivos.

Chamamos MEIO-SOMADOR (figura abaixo) ao circuito onde se pode somar apenas dois BITs, dando como resultado a soma S e o transporte T.



A	B	T	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Através da associação conveniente de meios-somadores, chegamos ao **somador completo,** que possibilita a realização de adição com números binários de diversos BITs.

- **Material a ser Utilizado**

CI's: 4030, 4081, 4072

OBSERVAÇÕES:

1. Se necessário usar combinações de portas, para conseguir a lógica de uma porta especifica;

2. Na montagem de um circuito lógico, implementar com CI's de uma mesma família (CMOS ou TTL).

- **Procedimentos**

1. Projete um somador completo capaz de somar dois números A e B, completando primeiramente o quadro abaixo e simplificando depois S e Ts por Karnaugh.

A	B	Ti	Ts	S
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Onde: **Ti** é o transporte de entrada

Ts é o transporte de saída s é o resultado da SOMA.

2. Implemente com outros 3 somadores completos um somador de 4 BITs.

10.7 Ensaio 7 (Flip-Flops)

● Objetivos

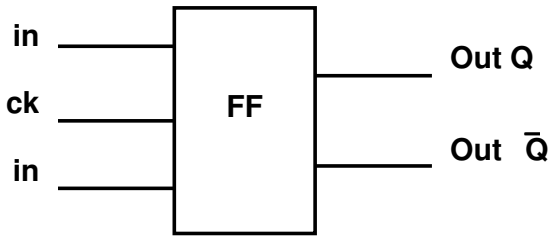
1. Montar Flip-Flops a partir de portas lógicas;
2. Conhecer Ci's contendo Flip-Flops;
3. Determinar as condições de saída dos diversos tipos de FF para diferentes combinações de estados na entrada.

● Informação Teórica

A lógica seqüencial é fundamental nas unidades de memória de uma posição (BIT) chamadas de Flip-Flop (FF) ou às vezes chamados de multivibradores.

O estado de um FF depende das variáveis de entrada e/ou de seus estados de saída anteriores, o que não ocorre com a lógica combinacional, na qual a saída só depende das variáveis de entrada.

A representação básica de um FF é dada na figura abaixo:



A entrada de controle, que é na realidade uma seqüência de pulsos, faz com que o FF mude seu estado de saída a cada pulso (clock).

Um FF tem a capacidade de armazenar um BIT por longos ou curtos intervalos de tempo. No primeiro caso, o seu emprego é em dispositivos de memória, como por exemplo, a RAM de microcomputadores, no segundo caso são utilizados como registradores ou contadores.

● **Material a ser Utilizado**

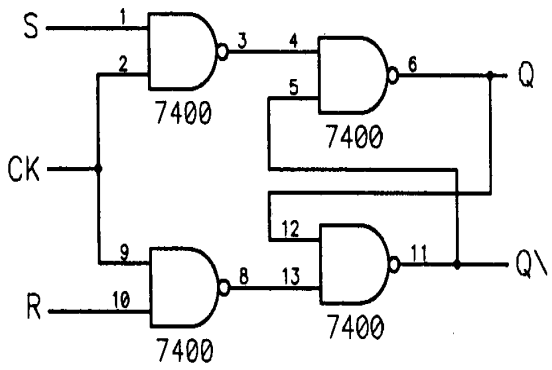
Cis: 7400, 4027

Fios para conexão

OBS.: Se necessário usar combinações de portas para conseguir a lógica de uma porta específica.

● **Procedimentos**

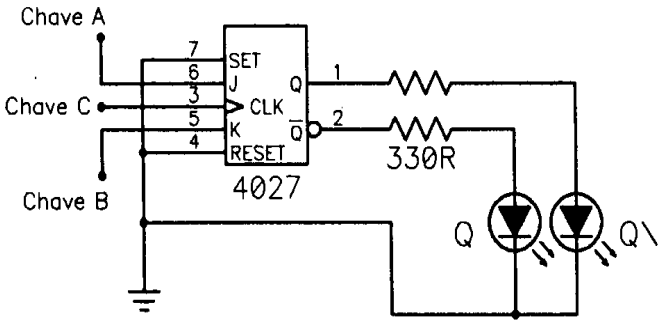
1. Monte o circuito lógico FF RS da figura abaixo e faça a sua tabela verdade.



Entradas			Saídas	
Ck	**S**	**R**	**Q**	**Q/**
1	0	0		
1	0	1		
1	1	0		
1	1	1		
0	X	X		

X – não importa

2. Usando o 4027, monte agora o circuito lógico do FF JK e determine Qf e Qf' na tabela verdade abaixo.



Entradas			Saídas	
Ck	**J**	**K**	**Q**	**Q**
↑	0	0		
↑	0	1		
↑	1	0		
↑	1	1		

↑ - transição (subida) do clock

3. Construa a partir de um FF JK um FF tipo D.

4. Construa a partir de um FF JK um FF tipo T.

10.8 Ensaio 8 (Contadores Assíncronos)

- **Objetivo**

Reconhecer o módulo de um contador.

- **Informação Teórica**

Um contador binário é um sistema seqüencial de uma entrada e várias saídas, que representam um número binário qualquer.

Basicamente os contadores são construídos por FF associados em série, levando-se cada saída dos FF às saídas do contador.

Normalmente a entrada de um contador é a própria entrada de clock de um FF ou mais da associação, e sendo assim, pode-se sensibilizar os FF induzindo-os a comutarem de maneira tal a incrementar o número da saída.

Com relação à sensibilização dos FF, os contadores podem ser classificados em SÍNCRONOS E ASSÍNCRONOS.

Os contadores assíncronos caracterizam-se por apresentarem a entrada no primeiro FF e a saída deste vai sensibilizar o seguinte.

Devido aos tempos de atraso inerente aos FF, as saídas não comutam ao mesmo tempo, ou seja, as comutações são defasadas (assíncronas).

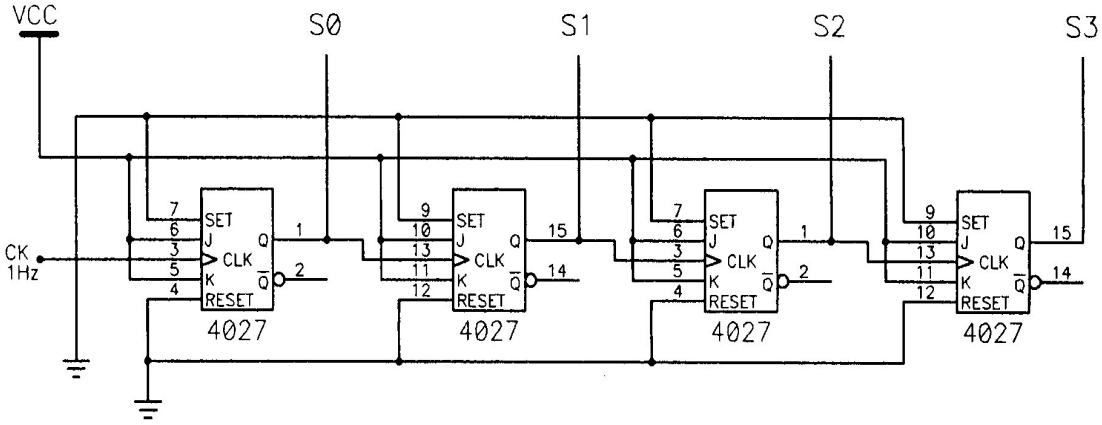
- **Material a ser Utilizado**

Ci's: 2 x 4027 e 4081

OBS.: Se necessário usar combinações de portas para conseguir a lógica de uma porta específica.

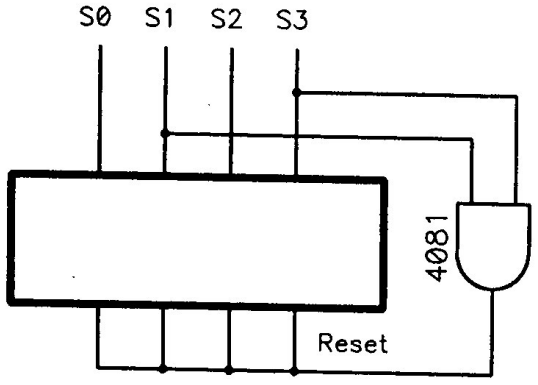
- **Procedimentos**

1. Monte o circuito lógico a seguir e complete a tabela verdade.



↑ Ck	S3	S2	S1	S0	Eq. Decimal
0					
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					

2. Complete o circuito do procedimento anterior de acordo com a figura abaixo.



10.9 Ensaio 9 (Contadores Síncronos)

● Objetivos

1. Montar um contador síncrono a partir de Flip-Flops;
2. Conhecer Cis contendo contadores.

● Informação Teórica

A segunda classe de contadores compreende os síncronos, ou seja, aqueles cujos FF comutam simultaneamente, não ocorrendo às combinações intermediárias indesejáveis durante as transições de estado.

Uma vantagem relevante deste contador sobre o assíncrono é quando a resposta de freqüência, devido ao problema dos acúmulos de atraso dos FF, podendo trabalhar com freqüência mais elevadas.

Além disso, nos contadores síncronos não nos limitamos a poder gerar uma seqüência crescente ou decrescente, podemos gerar qualquer seqüência de números que desejarmos.

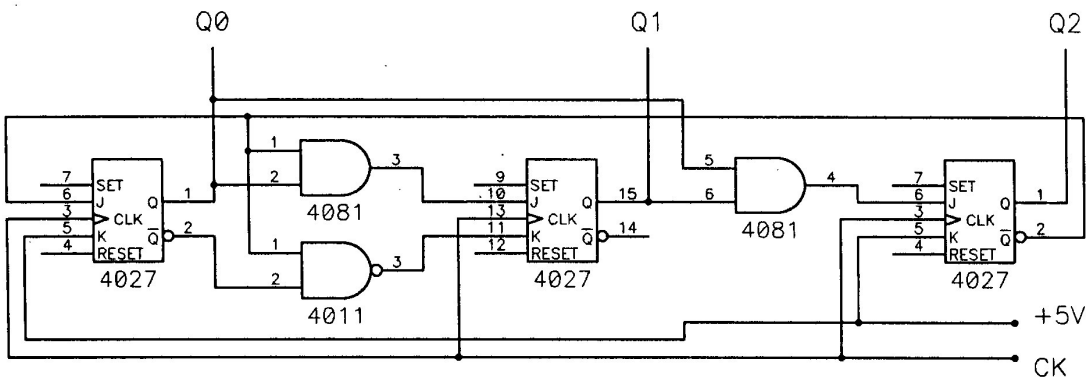
Referente aos contadores síncronos, o FF JK é bastante utilizado, por isso tenha sempre em mente sua tabela verdade.

J	K	Qf
0	0	Qa
0	1	0
1	0	1
1	1	$\overline{Q}$a

● Material a ser Utilizado

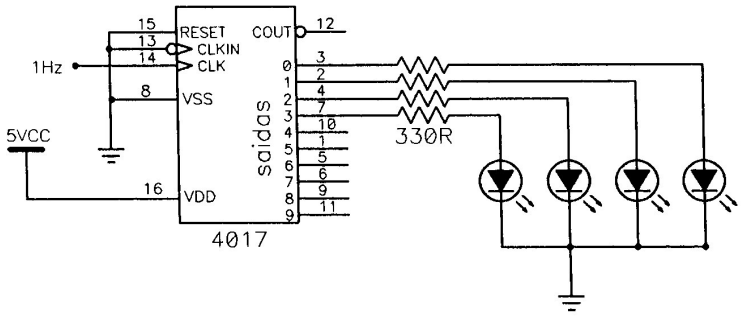
Cis: 2 x 4027, 4011, 4081, 4017 e 4510
Fios para conexão

OBS.: Se necessário usar combinações de portas para conseguir a lógica de uma porta específica.



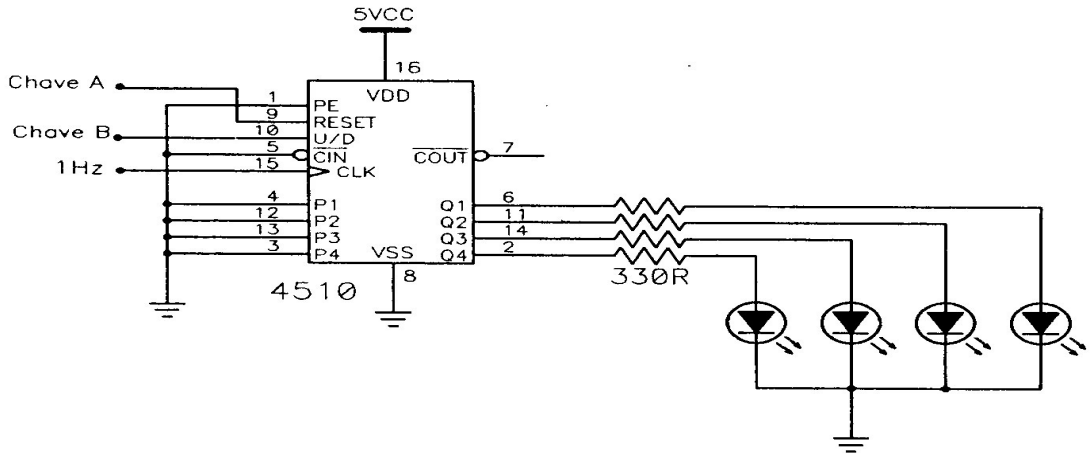
Qual o módulo do contador acima? Por que?

2. Utilizando agora o 4017, monte o circuito abaixo. **Alimentação V_{CC} = 5V/GND.** Como se comporta o circuito?



3. Usando o 4510, monte o circuito e complete o quadro abaixo:

Entradas		Saídas			
Chave A	Chave B	Q3	Q2	Q1	Q0



4. utilizando o 4510 e display de 7 segmentos, elabore um contador de 0 – 9.

5. Utilizando os componentes citados acima elabore um relógio digital.

10.10 Ensaio 10 (Multiplex/Demultiplex)

● Objetivos

1. Montar multiplexadores a partir de portas lógicas;
2. Conhecer Cis multiplexadores/demultiplexadores.

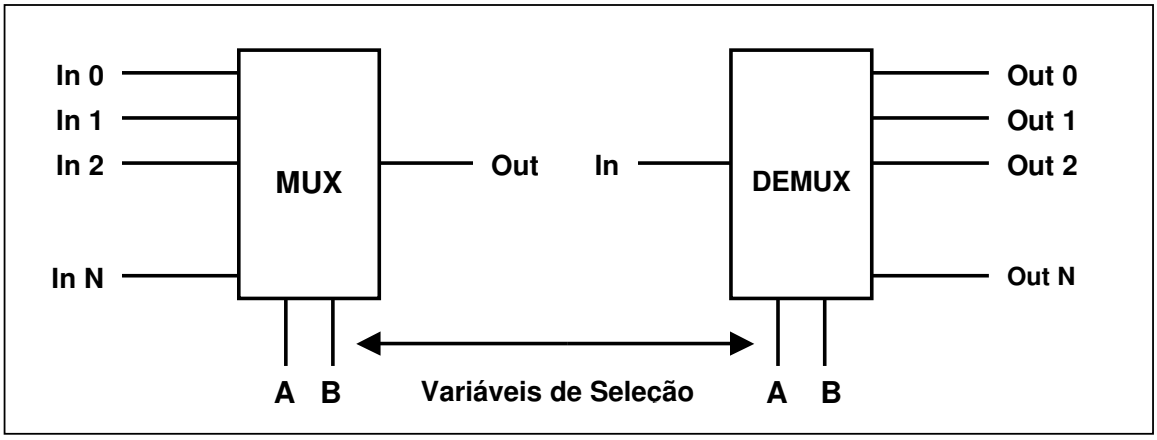
● Informação Teórica

Os circuitos multiplex são utilizados quando necessitamos enviar um determinado número de informações contidas em vários canais por um único canal.

Já os circuitos demultiplex efetuam a função inversa dos multiplex, ou seja, enviam as informações recebidas de um único canal em vários canais.

Estes dois tipos de circuitos são largamente empregados em transmissão de dados, em telefonia ou em outros circuitos que requerem esta função.

O quadro abaixo mostra em forma de blocos estes dois tipos de circuitos.



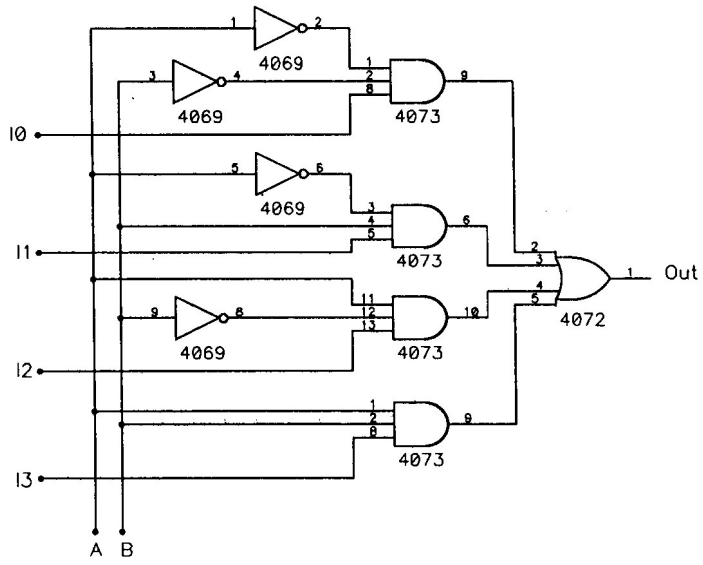
● **Material a ser Utilizado**

CI's: 4069, 4072, 4073, 74153 e 74155.

Fios para conexão

● **Procedimentos**

1. Monte o circuito lógico da figura. **Alimentação V_{CC} = 5V/GND**



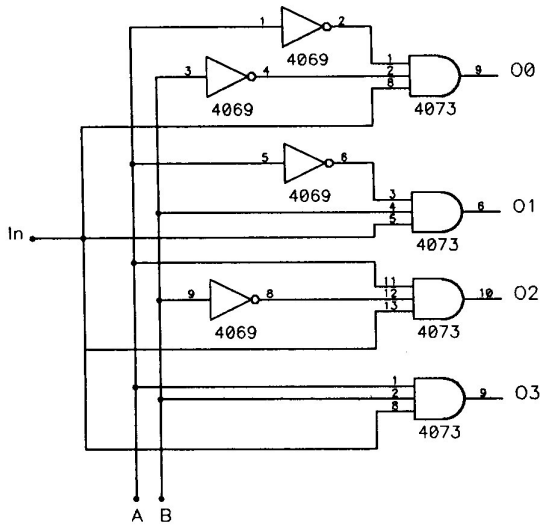
2. Faça Ino e In2 permanecer em nível 1, In1 em nível 0.

3. Coloque A e B em duas chaves de dados quaisquer e OUT em um indicador lógico, em seguida complete a tabela abaixo.

A	B	Out
0	0	
0	1	
1	0	
1	1	

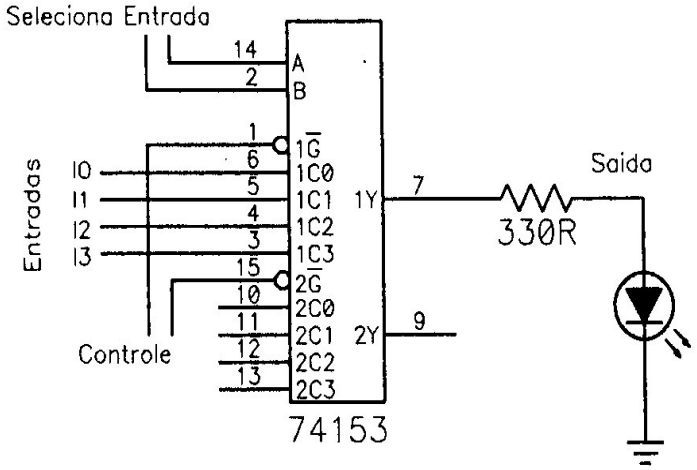
4. Modifique o estado das entradas e observe a saída (OUT) quando se varia A e B. Comente.

5. Monte o circuito abaixo apresentado e complete a sua tabela verdade.



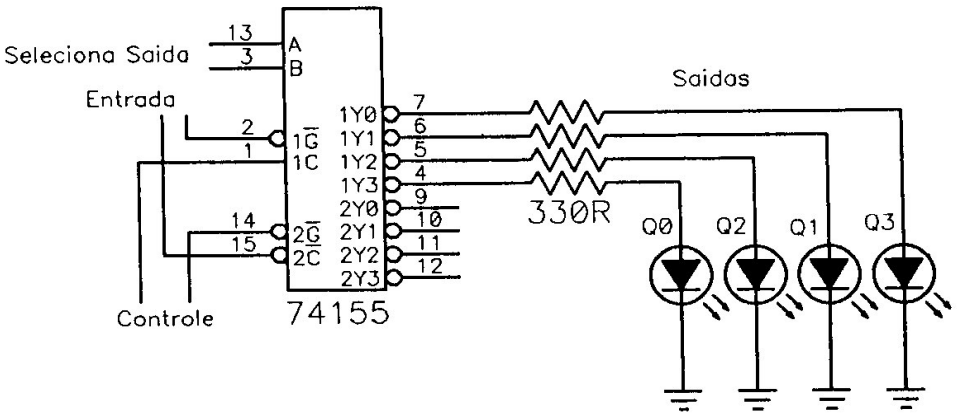
In	A	B	Q0	Q1	Q2	Q3
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

6. Monte um circuito lógico com 74153 (mux) que possibilite e o preenchimento da tabela abaixo. **Alimentação V_{CC} = 5V/GND**



Seleção		Entradas de Dados				Controle		Saída de Dados

7. Monte um circuito lógico com 74155 (demux) que possibilite o preenchimento da tabela abaixo. **Alimentação V_{CC} = 5V/GND**



seleção		Entrada de dado	Controle		Saída de Dados			

10.11 Ensaio 11 (Registradores de Deslocamento)

● Objetivos

1. Conhecer o funcionamento básico de Registrador de Deslocamento;
2. Montar circuitos com registradores de deslocamento em uma única pastilha.

● Informação Teórica

O registrador de deslocamento constitui um dos dispositivos mais usados em sistemas digitais. Numa simples calculadora temos exemplo das características do registrador de deslocamento.

Para se introduzir o número 47 na calculadora, a tecla 4 é pressionada e depois liberada. O número 4 aparece no display. A tecla 7 é então pressionada e liberada. O número 47 é apresentado no display. Ocorreu o deslocamento da posição inicial do número 4 no display.

Uma outra característica, além do deslocamento, a calculadora apresenta a capacidade de armazenar os números introduzidos. Quando o número 4 foi introduzido e a tecla liberada, o registrador ainda memorizou esta tecla pressionada. Esta função é de grande importância para muitos circuitos digitais.

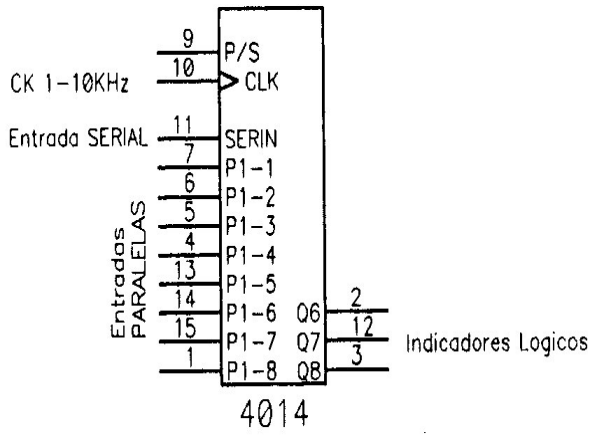
Os registradores de deslocamento são usados como memória temporária e para deslocamento de dados à esquerda ou à direita, assim como também para transformar dados em série para paralelo e vice-versa.

● Material a ser Utilizado

Cis: 4014 e 4015
Fios para conexão.

● **Procedimentos**

1. Monte o circuito a seguir observando com cuidado a perfeita localização de cada entrada/saída. **Alimentação V_{CC} = 5V/GND**



2. Armazene o dado I = 10010100 serialmente e depois paralelamente, selecionando o modo de entrada pelo pino 9 (PE). Observe as saídas através dos indicadores lógicos e comprove se a informação foi introduzida corretamente.

3. Utilizando o 4015, projete um registrador de deslocamento de 8 bits para entrada serial e saída paralela.

11 DIGITAL INTEGRATED CIRCUITS

11.1 Function Selection Chart

Function	Type No.	No. of Pins
Gates		
NOR/NAND		
Dual 4-input NOR	CD4002B	14
	CD4002UB	14
	CD4002A	14
Dual 4-input NAND	CD4012B	14
	CD4012UB	14
	CD4012A	14
Triple 3-input NOR	CD4025B	14
	CD4025UB	14
	CD4025A	14
Triple 3-input NAND	CD4023B	14
	CD4023UB	14
	CD4023A	14
Quad 2-input NOR	CD4001B	14
	CD4001UB	14
	CD4001A	14
Quad 2-input NAND	CD4011B	14
	CD4011UB	14
	CD4011A	14
8-input NOR/OR	CD4078B	14
8-input NAND/AND	CD4068B	14
Dual 3-input NOR plus inverter	CD4000B	14
	CD4000UB	14
	CD4000A	14
Dual 2-input NAND buffer/driver	CD40107B	8,14
OR/AND		
Dual 4-input OR	CD4072B	14
Dual 4-input AND	CD4082B	14
Triple 3-input OR	CD4075B	14
Triple 3-input AND	CD4073B	14
Quad 2-input OR	CD4071B	14
Quad 2-input AND	CD4081B	14
Buffers and Inverters		
Dual complementary pair plus inverter	CD4007UB	14
	CD4007A	14
Hex inverter	CD4069UB	14
Strobed hex inverter/buffer	CD4502B	16
Hex buffer/converter (inverting)	CD4009UB	16
	CD4009A	16
Hex buffer/converter (inverting)	CD4049UB	16
	CD4049A	16
Hex buffer/converter (non-inverting)	CD4010B	16
	CD4010A	16
Hex buffer/converter (non-inverting)	CD4050B	16
	CD4050A	16
Quad true/complement buffer	CD4041UB	14
	CD4041A	14
Dual 2-input NAND buffer/driver	CD40107B	8,14

Function	Type No.	No. of Pins
Gates (cont'd)		
Multifunction/AOI		
Triple AND-OR bi-phase pairs	CD4037A	14
Quad exclusive-OR	CD4030B	14
	CD4030A	14
Quad exclusive-OR	CD4070B	14
Quad exclusive-NOR	CD4077B	14
Quad AND/OR Select	CD4019B	16
	CD4019A	16
Dual 2-wide, 2-input AND/OR invert (AOI)	CD4085B	14
Expandable 4-wide, 2-input AND/OR invert (AOI)	CD4086B	14
Multifunctional expandable 8-input (3 state output)	CD4048B	16
	CD4048A	16
Decoders/Encoders		
BCD-to-decimal decoder	CD4028B	16
	CD4028A	16
8-input priority encoder	CD4532B	16
10-line to 4-line BCD priority encoder	CD40147B	16
4-bit latch/4-to-16 line decoder (outputs high)	CD4514B	24
4-bit latch/4-to-16 line decoder (outputs low)	CD4515B	24
Dual 1-of-4 decoder/demultiplexer (outputs high)	CD4555B	16
Dual 1-of-4 decoder/demultiplexer (outputs low)	CD4556B	16
Schmitt Trigger		
Quad 2-input NAND	CD4093B	14
Hex	CD40106B	14
Interface		
Quad low-to-high voltage	CD40109B	16
Hex high-to-low voltage (inverting)	CD4009UB	16
	CD4009A	16
	CD4049UB	16
	CD4049A	16
Hex high-to-low voltage (non-inverting)	CD4010B	16
	CD4010A	16
	CD4050B	16
	CD4050A	16
Dual 2-input NAND buffer/driver	CD40107B	8,14
8-bit bidirectional	CD40115▼	22
Multivibrators		
Monostable/astable	CD4047B	14
	CD4047A	14
Dual monostable	CD4098B	16

▼ Indicates types designed for special applications. Ratings and characteristics data for these types differ in some aspects from the standardized data for. A- and B- series types. Refer to RCA data bulletin on these types for specific differences. Data bulletin file numbers are shown on functional diagrams.

Function Selection Chart

Function	Type No.	No. of Pins
Multivibrators (cont'd)		
Flip-Flops		
Dual "D" with set/reset capability	**CD4013B**	14
	CD4013A	14
Dual "J-K" with set/reset capability	**CD4027B**	16
	CD4027A	16
Gated "J-K" (non-inverting)	**CD4095B**	14
Gated "J-K" (inverting and non-inverting)	**CD4096B**	14
Hex "D"	**CD40174B**	16
4-bit "D" with 3-state outputs	**CD4076B**	14
Latches		
Quad clocked "D"	**CD4042B**	16
	CD4042A	16
Quad NOR R/S (3-state outputs)	**CD4043B**	16
	CD4043A	16
Quad NAND R/S (3-state outputs)	**CD4044B**	16
	CD4044A	16
Dual 4-bit	**CD4508B**	24
8-bit addressable	**CD4099B**	16
	CD4724B	16
Registers		
Shift Registers-Static		
Dual 4-stage with serial input/parallel output	**CD4015B**	16
	CD4015A	16
18-stage	**CD4006B**	14
	CD4006A	14
64-stage	**CD4031B**	16
	CD4031A	16
Dual 64-bit	**CD4517B**	16
8-stage with synchronous parallel or serial input/serial output	**CD4014B**	16
	CD4014A	16
8-stage with asynchronous parallel input or synchronous serial input/serial output	**CD4021B**	16
	CD4021A	16
4-stage parallel-in/parallel-out with J-K input and true/complement output	**CD4035B**	16
	CD4035A	16
8-stage bidirectional parallel or serial input/parallel output	**CD4034B**	24
	CD4034A	24
32-bit left/right	**CD40100B**	16
8-stage shift-and-store bus	**CD4094B**	16

Function	Type No.	No. of Pins
Registers (cont'd)		
Storage Registers		
8-bit addressable latch	**CD4099B**	16
	CD4724B	16
4-bit "D"-type with 3-state outputs	**CD4076B**	16
4 X 4 Multiport	**CD40108B**	24
4 X 4 Multiport	**CD40208B**	24
FIFO Buffer Registers		
4-bit X 16 word	**CD40105B**	16
Counters		
Binary Ripple		
7-stage	**CD4024B**	14
	CD4024A	14
12-stage	**CD4040B**	16
	CD4040A	16
14-stage	**CD4020B**	16
	CD4020A	16
14-stage counter/divider and oscillator	**CD4060B**	16
	CD4060A	16
Timers		
21-stage	**CD4045B**	14
	CD4045A	14
Programmable	**CD4536B**	16
Synchronous		
Decade counter/divider plus 10 decoded decimal outputs	**CD4017B**	16
	CD4017A	16
Divide-by-8 counter/divider with 8 decimal outputs	**CD4022B**	16
	CD4022A	16
Presettable divide-by-"N" counter, fixed or programmable	**CD4018B**	16
	CD4018A	16
Programmable divide-by-"N" counter	**CD4059A**	24
Presettable up/down counter, binary or BCD-decade	**CD4029B**	16
	CD4029A	16
Presettable 4-bit BCD up/down counter	**CD4510B**	16
Presettable 4-bit binary up/down counter	**CD4516B**	16
Presettable 2-decade BCD down counter	**CD40102B**	16
Presettable 8-bit binary down counter	**CD40103B**	16
Presettable 4-bit BCD up/down counter	**CD40192B**	16

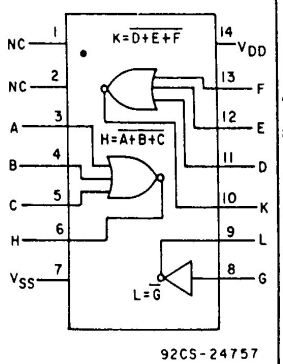
Function Seletion Chart

Function	Type No.	No. of Pins
Counters (cont'd)		
Synchronous (cont'd)		
Presettable 4-bit binary up/down counter	CD40193B	16
Dual BCD up counter	CD4518B	16
Dual binary up counter	CD4520B	16
Decade counter/asynchronous clear	CD40160B	16
Binary counter/asynchronous clear	CD40161B	16
Decade counter/synchronous clear	CD40162B	16
Binary counter/synchronous clear	CD40163B	16
Display Drivers		
With Counter		
Decade counter/divider with 7-segment display outputs and display enable	CD4026B	16
	CD4026A	16
Decade counter/divider with 7-segment display outputs and ripple blanking	CD4033B	16
	CD4033A	16
Up/Down Counter-Latch-Decoder-Driver	CD40110B	16
For Liquid-Crystal-Display Drive		
4-line BCD-to-7-segment decoder/driver	CD4054B	16
with "display-frequency" output	CD4055B	16
BCD-to-7-segment decoder/driver with strobed-latch function	CD4056B	16
For Light-Emitting-Diode Drive		
BCD-to-7-segment latch decoder/driver	CD4511B	16
Multiplexers/Demultiplexers		
Analog/Digital		
Triple 2-channel	CD4053B	16
Differential 4-channel	CD4052B	16
Single 8-channel	CD4051B	16
Differential 8-channel	CD4097B	24
Single 16-channel	CD4067B	24
Quad bilateral switch	CD4016B	14
	CD4016A	14
Quad bilateral switch	CD4066B	14
	CD4066A	14
Quad AND/OR select	CD4019B	16
	CD4019A	16
Dual 1-of-4 decoder/demultiplexer (outputs high)	CD4555B	16
Dual 1-of-4 decoder/demultiplexer (outputs low)	CD4556B	16

Function	Type No.	No. of Pins
Multiplexers/Demultiplexers (cont'd)		
Data Selector		
Quad 2-line-to-1-line	CD40257B	16
Phase Locked Loop		
Micropower	CD4046B	16
	CD4046A	16
Arithmetic Circuits		
Adders/Comparators		
4-bit full adder with parallel carry out	CD4008B	16
	CD4008A	16
Triple serial adder, positive logic	CD4032B	16
	CD4032A	16
Triple serial adder, negative logic	CD4038B	16
	CD4038A	16
4-bit magnitude comparator	CD4063B	16
	CD4585B	16
Quad exclusive-OR gate	CD4030B	14
	CD4030A	14
Quad exclusive-OR gate	CD4070B	14
Quad exclusive-NOR gate	CD4077B	14
ALU/Rate Multipliers		
4-bit arithmetic logic unit	CD40181B	24
	CD4057A	28
BCD rate multiplier	CD4527B	16
Binary rate multiplier	CD4089B	16
Look-ahead-carry block	CD40182B	16
Parity Generator/Checker		
9-bit	CD40101B	14
Multiport Register		
4 X 4	CD40108B	24
4 X 4	CD40208B	24
Quad Bilateral Switches		
For transmission or multiplexing of analog or digital signals	CD4016B	14
	CD4016A	14
	CD4066B	14
	CD4066A	14
Crosspoint Switches		
4 X 4 crosspoint switch with control memory	CD22100▼	16
4 X 4 X 2 crosspoint switch with control memory	CD22101▼	24
4 X 4 X 2 crosspoint switch with control memory	CD22102▼	24
Memory		
16 X 4 static RAM	CD40114B	16

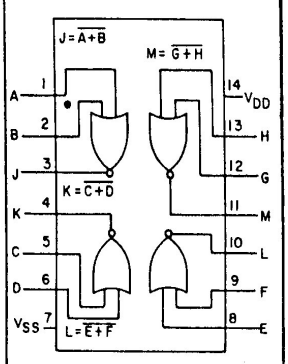
▼ Indicates types designed for special applications. Ratings and characteristics data for these types differ in some aspects from the standardized data for. A- and B- series types. Refer to RCA data bulletin on these types for specific differences. Data bulletin file numbers are shown on functional diagrams.

11.2 Functional Diagrams



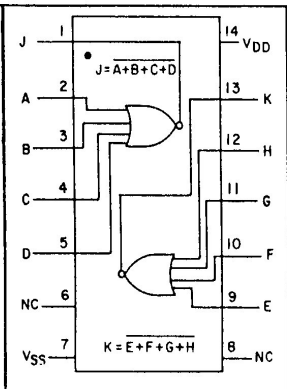
Dual 3-Input NOR Gate Plus Inverter

CD4000A	(File No. 944)
CD4000B	(File No. 985)
CD4000UB	(File No. 945)



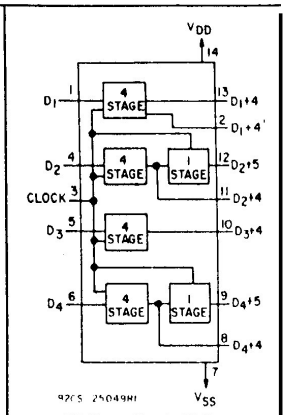
Quad 2-Input NOR Gate

CD4001A	(File No. 944)
CD4001B	(File No. 985)
CD4001UB	(File No. 945)



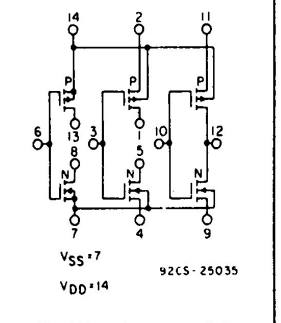
Quad 4-Input NOR Gate

CD4002A	(File No. 944)
CD4002B	(File No. 985)
CD4002UB	(File No. 945)



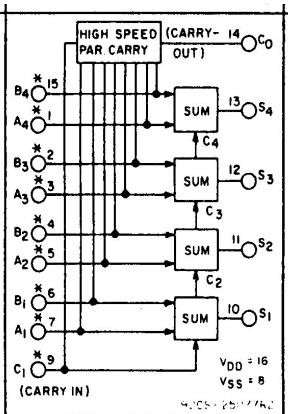
18-Stage Static Shift Register

CD4006A	(File No. 920)
CD4006B	(File No. 1033)



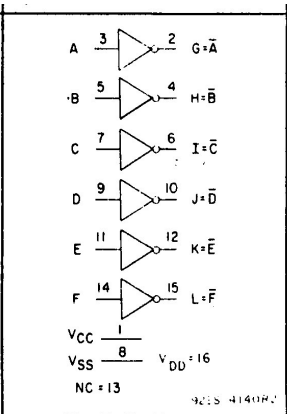
Dual Complementary Pair Plus Inverter

CD4007A	(File No. 921)
CD4007UB	(File No. 977)



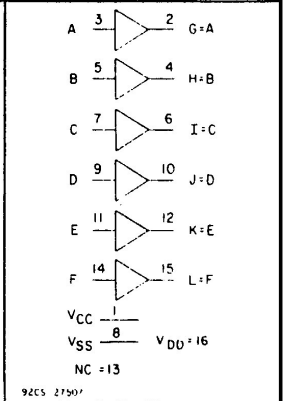
4-Bit Full Adder with Parallel Carry Out

CD4008A	(File No. 950)
CD4008B	(File No. 951)



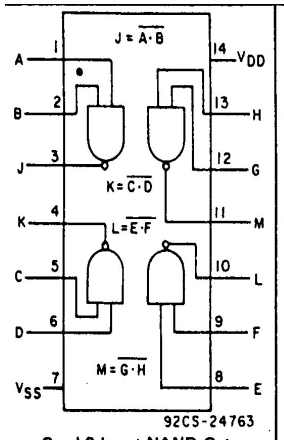
Hex Buffer/Converter Inverting Type

CD4009A	(File No. 939)
CD4009UB	(File No. 940)



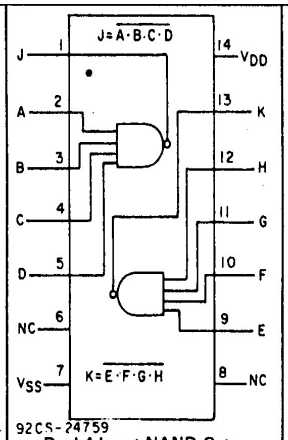
Hex Buffer/Converter Non-Inverting Type

CD4010A	(File No. 939)
CD4010B	(File No. 940)



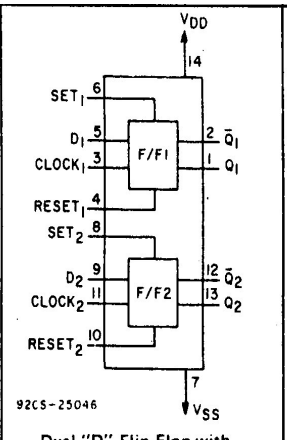
Quad 2-Input NAND Gate

CD4011A	(File No. 946)
CD4011B	(File No. 986)
CD4011UB	(File No. 947)



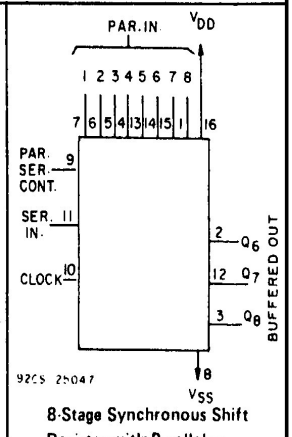
Dual 4-Input NAND Gate

CD4012A	(File No. 946)
CD4012B	(File No. 986)
CD4012UB	(File No. 947)



Dual "D" Flip-Flop with Set/Reset Capability

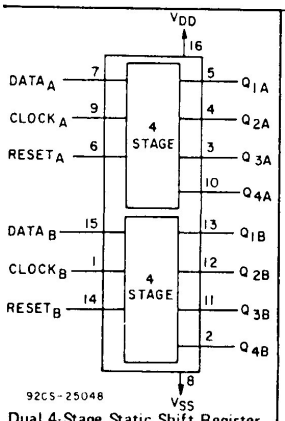
CD4013A	(File No. 935)
CD4013B	(File No. 936)



8-Stage Synchronous Shift Register with Parallel or Serial Input/Serial Output

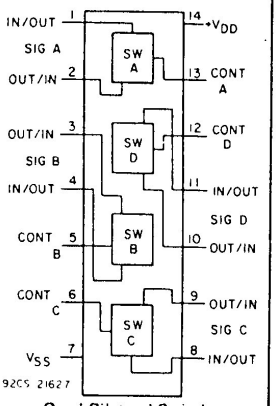
CD4014A	(File No. 922)
CD4014B	(File No. 1043)

Functional Diagrams



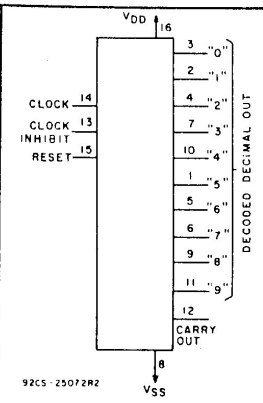
Dual 4-Stage Static Shift Register with Serial Input/Parallel Output

CD4015A	(File No. 943)
CD4015B	(File No. 1024)



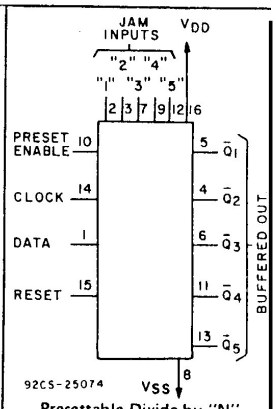
Quad Bilateral Switch

CD4016A	(File No. 952)
CD4016B	(File No. 953)



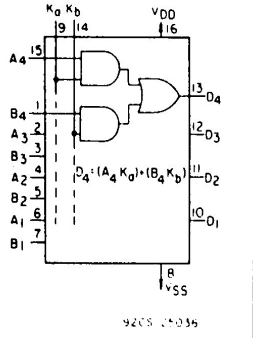
Decade Counter/Divider with 10 Decoded Decimal Outputs

CD4017A	(File No. 927)
CD4017B	(Prel.)



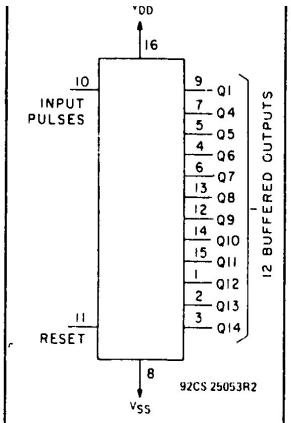
Presettable Divide-by-"N" Counter Fixed or Programmable

CD4018A	(File No. 929)
CD4018B	(File No. 1034)



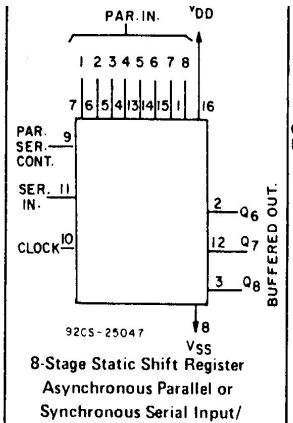
Quad AND/OR Select Gate

CD4019A	(File No. 923)
CD4019B	(File No. 1045)



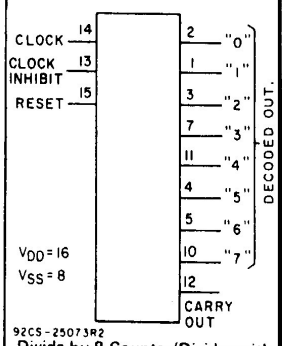
14-Stage Binary Ripple Counter

CD4020A	(File No. 928)
CD4020B	(File No. 1063)



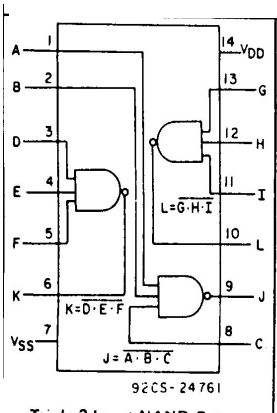
8-Stage Static Shift Register Asynchronous Parallel or Synchronous Serial Input/ Serial Output

CD4021A	(File No. 933)
CD4021B	(File No. 1043)



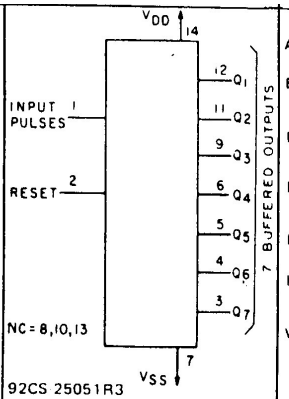
Divide-by-8 Counter/Divider with 8 Decoded Decimal Outputs

CD4022A	(File No. 919)
CD4022B	(Prel.)



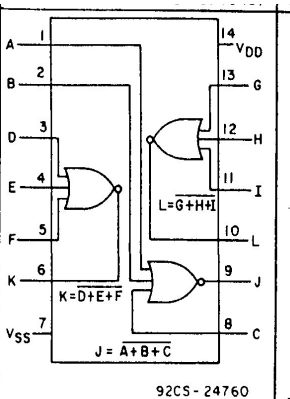
Triple 3-Input NAND Gate

CD4023A	(File No. 946)
CD4023B	(File No. 986)
CD4023UB	(File No. 947)



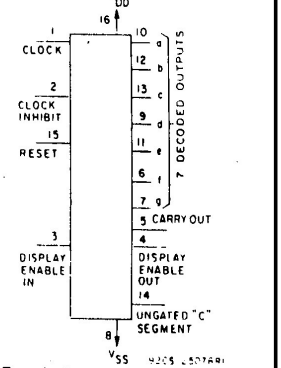
7-Stage Ripple-Carry Binary Counter/Divider

CD4024A	(File No. 930)
CD4024B	(File No. 1063)



Triple 3-Input NOR Gate

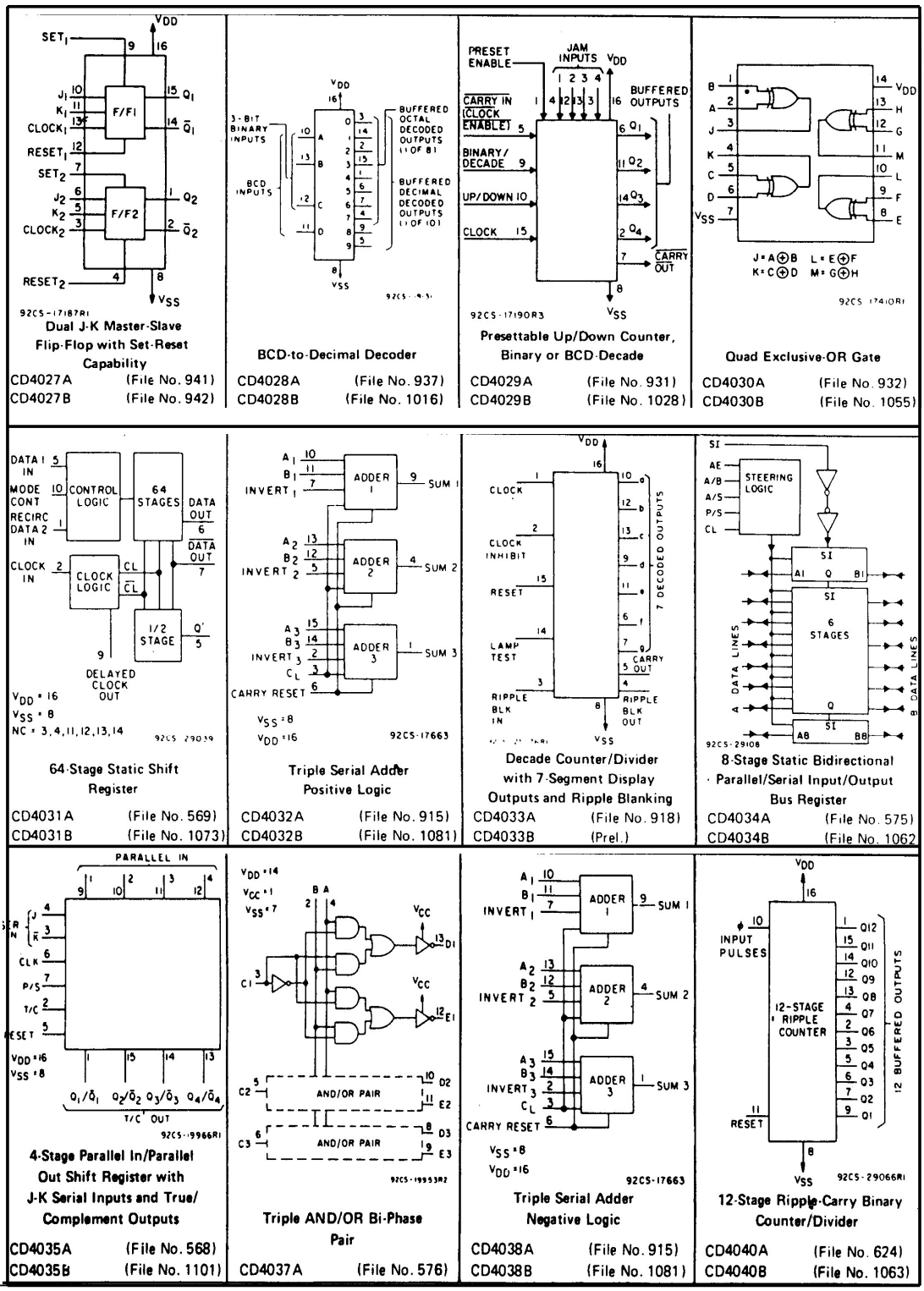
CD4025A	(File No. 944)
CD4025B	(File No. 985)
CD4025UB	(File No. 945)



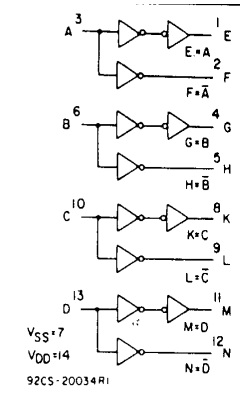
Decade Counter/Divider with 7-Segment Display Outputs and Display Enable

CD4026A	(File No. 918)
CD4026B	(Prel.)

Functional Diagrams

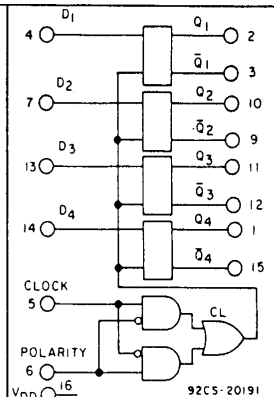


Functional Diagrams



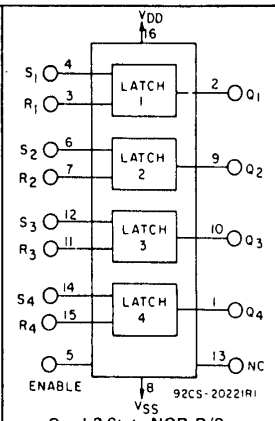
Quad True/Complement Buffer

CD4041A (File No. 572)
CD4041UB (File No. 934)



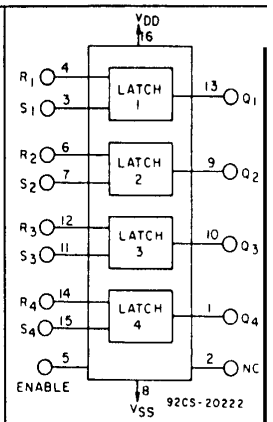
Quad Clocked "D" Latch

CD4042A (File No. 589)
CD4042B (File No. 954)



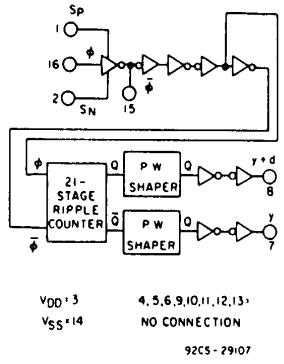
Quad 3-State NOR R/S Latch

CD4043A (File No. 590)
CD4043B (File No. 956)



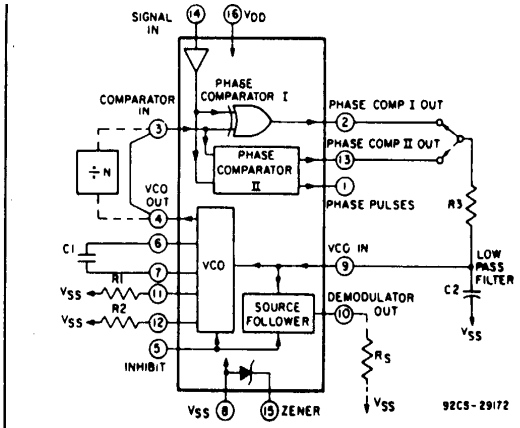
Quad 3-State NAND R/S Latch

CD4044A (File No. 590)
CD4044B (File No. 956)



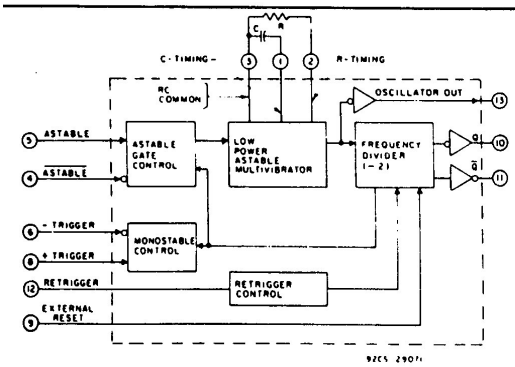
21-Stage Counter

CD4045A (File No. 614)
CD4045B (Prel.)



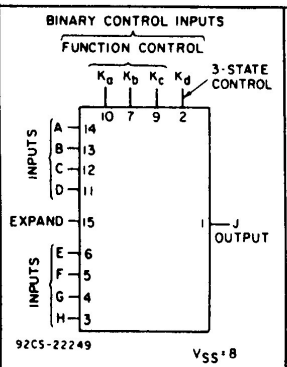
Micropower Phase-Locked Loop

CD4046A (File No. 637)
CD4046B (File No. 1099)



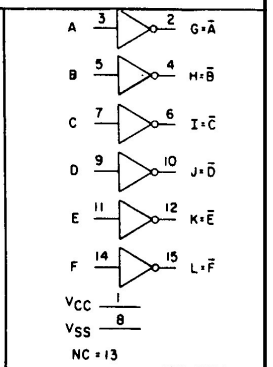
Low-Power Monostable/Astable Multivibrator

CD4047A (File No. 623)
CD4047B (Prel.)



Multi-Function Expandable 8-Input Gate

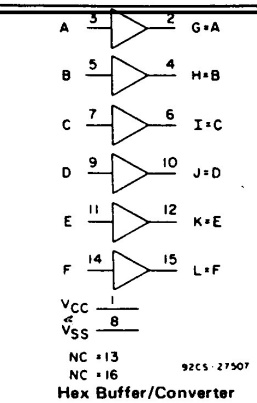
CD4048A (File No. 636)
CD4048B (Prel.)



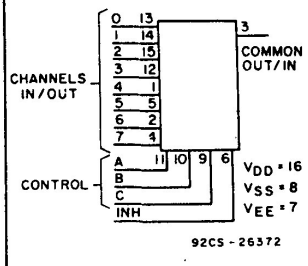
Hex Buffer/Converter Inverting Type

CD4049A (File No. 599)
CD4049UB (File No. 926)

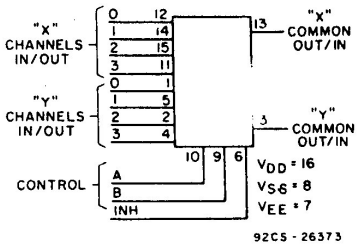
Functional Diagrams



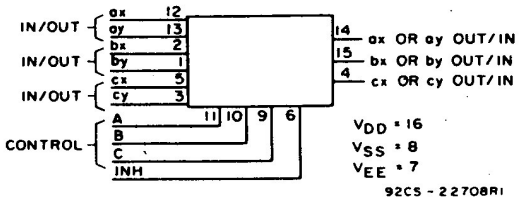
Hex Buffer/Converter
Non-Inverting Type
CD4050A (File No. 599)
CD4050B (File No. 926)



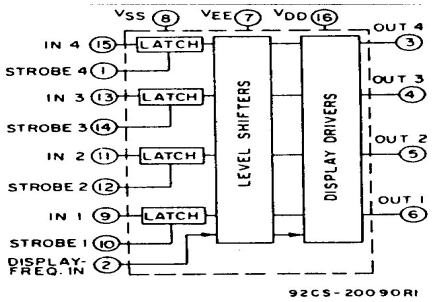
Single 8-Channel Analog
Multiplexer/Demultiplexer
CD4051B (File No. 902)



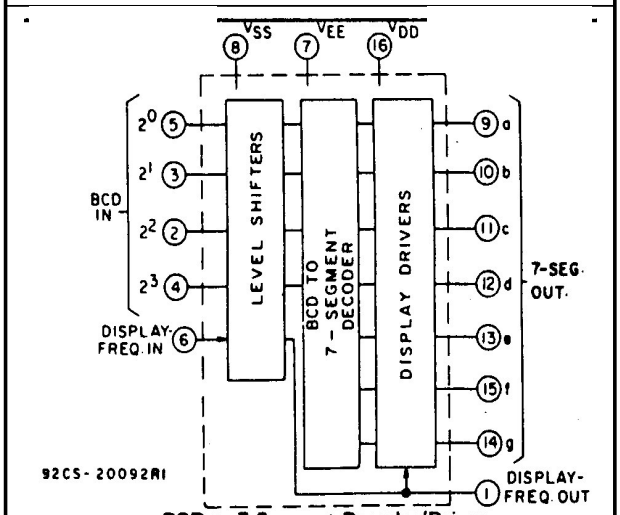
Differential 4-Channel Analog
Multiplexer/Demultiplexer
CD4052B (File No. 902)



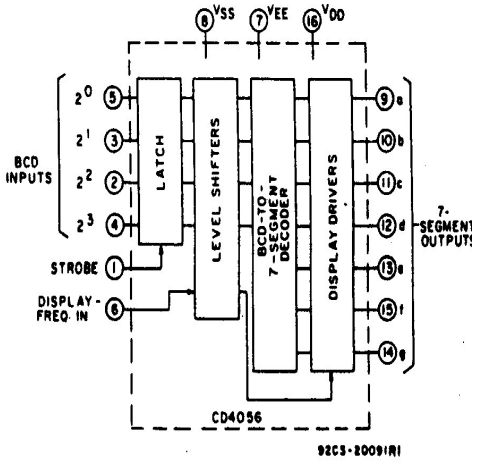
Triple 2-Channel
Multiplexer/Demultiplexer



4-Segment Liquid-Crystal
Display Driver
CD4054B (File No. 634)

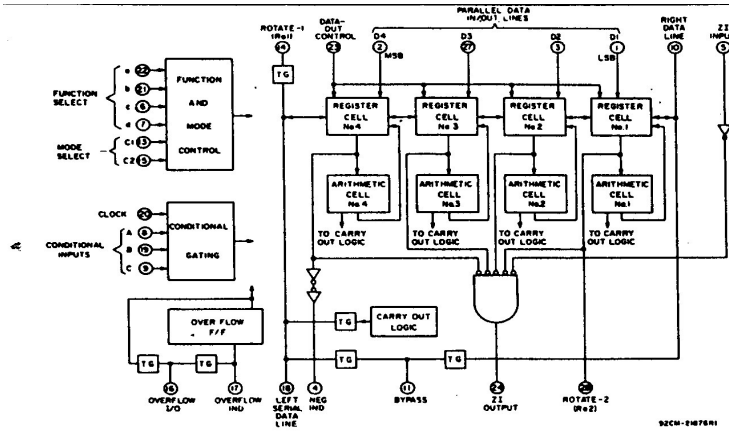


BCD-to-7-Segment Decoder/Driver
with "Display-Frequency" Output
Liquid-Crystal Display Driver



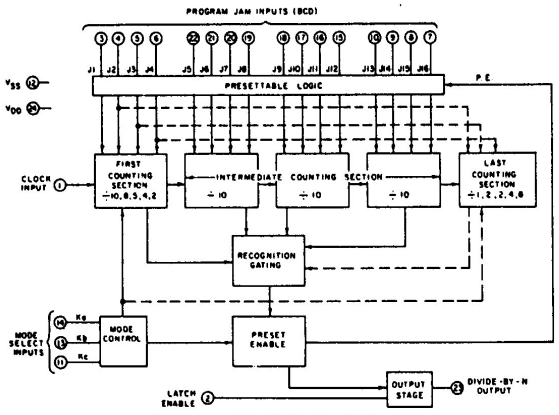
BCD-to-7-Segment Decoder/Driver
with Strobed-Latch Function
Liquid-Crystal Display Driver
CD4056B (File No. 634)

Functional Diagrams



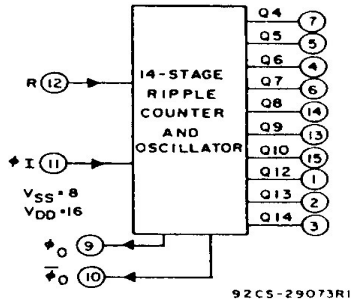
4-Bit Arithmetic Logic Unit

CD4057A (File No. 635)



Programmable Divide-by-"N" Counter

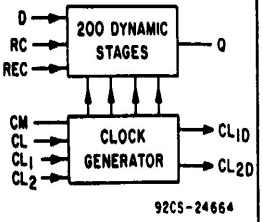
CD4059A (File No. 898)



14-Stage Ripple-Carry Binary Counter/Divider and Oscillator

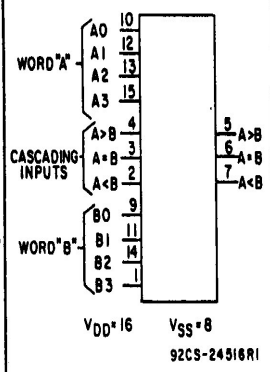
CD4060A (File No. 813)

CD4060B (Prel.)



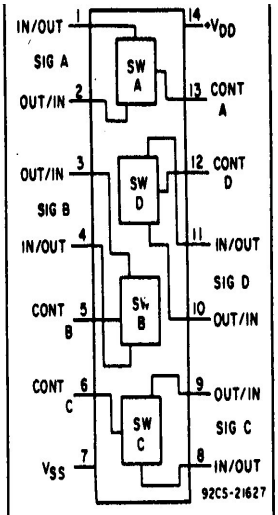
200-Stage Dynamic Shift Register

CD4062A (File No. 816)



4-Bit Magnitude Comparator

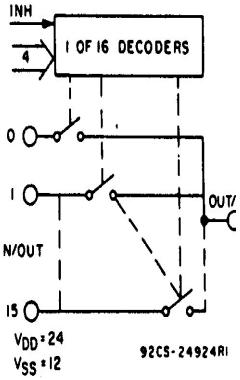
CD40638 (File No. 805)



Quad Bilateral Switch

CD4066A (File No. 769)

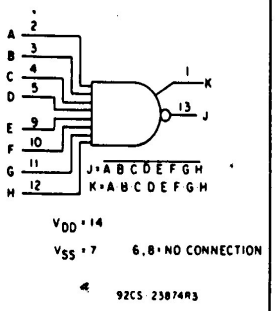
CD4066B (Prel.)



16-Channel Multiplexer/Demultiplexer

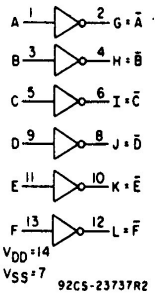
CD4067B (File No. 909

Functional Diagrams



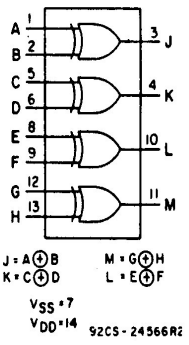
8-Input NAND/AND Gate

CD4068B (File No. 809)



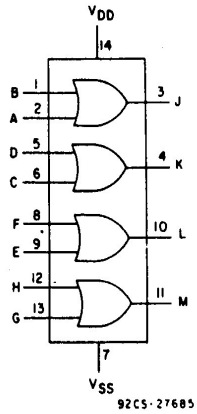
Hex Inverter

CD4069UB (File No. 804)



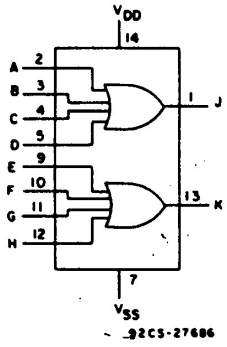
Quad Exclusive-OR Gate

CD4070B (File No. 910)



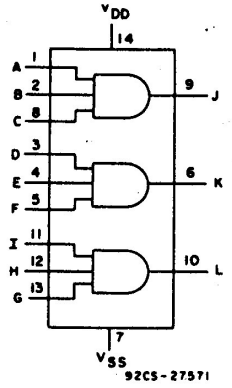
Quad 2-Input OR Gate

CD4071B (File No. 807)



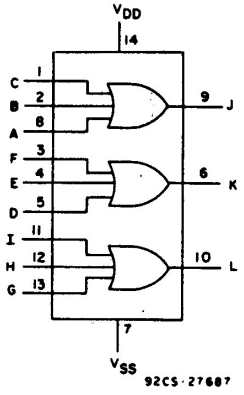
Dual 4-Input OR Gate

CD4072B (File No. 807)



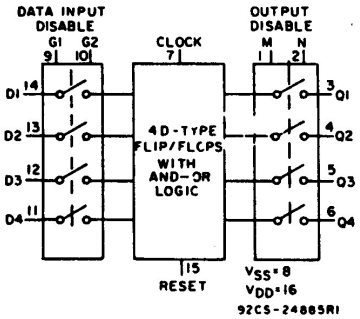
Triple 3-Input AND Gate

CD4073B (File No. 806)



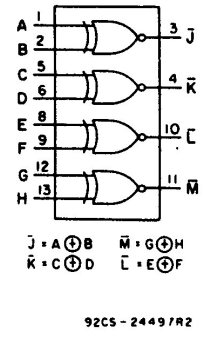
Triple 3-Input OR Gate

CD4075B (File No. 807)



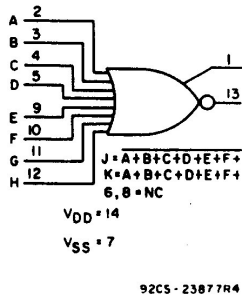
4-Bit D-Type Register

CD4076B (File No. 903)



Quad Exclusive-NOR Gate

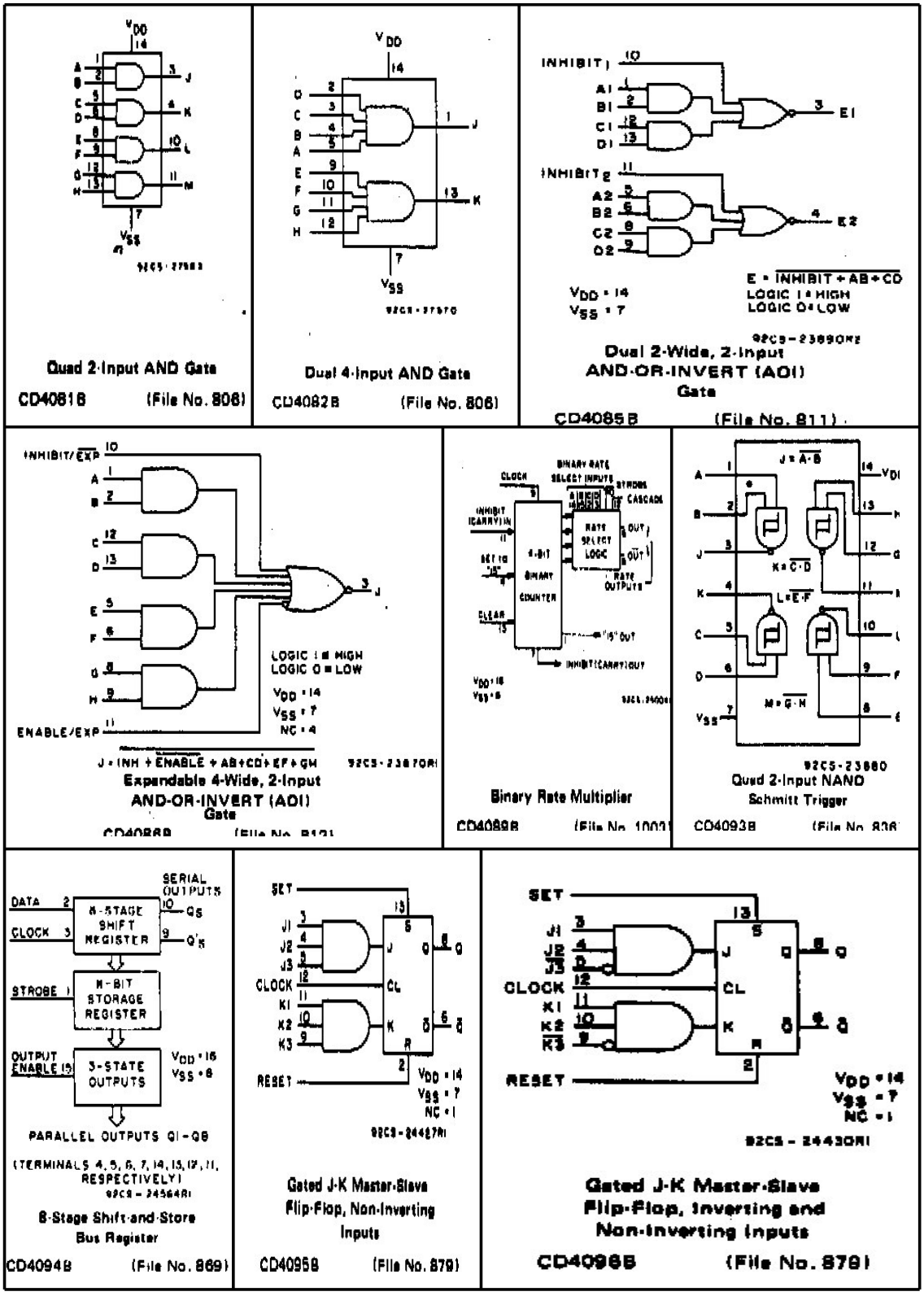
D4077B (File No. 910)



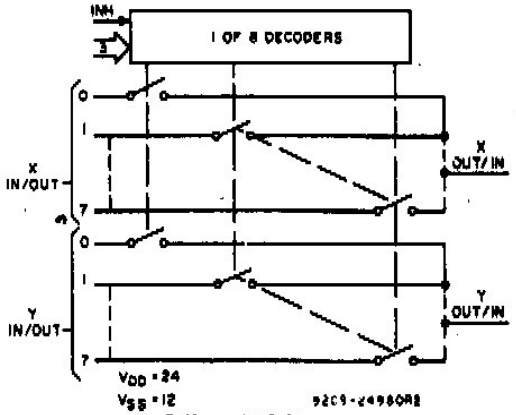
8-Input NOR/OR Gate

CD4078B (File No. 8

Functional Diagrams

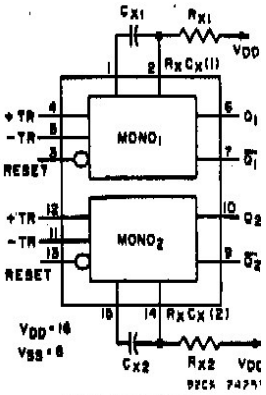


Functional Diagrams



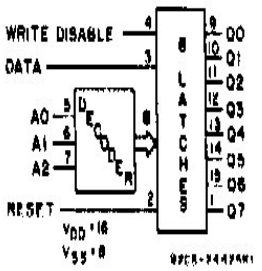
Differential 8-Channel Multiplexer/Demultiplexer

CD4097B (File No. 909)



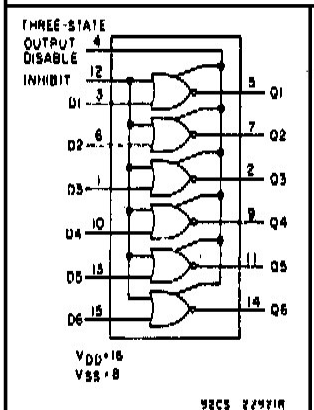
Dual Monostable Multivibrator

CD4098B (File No. 979)



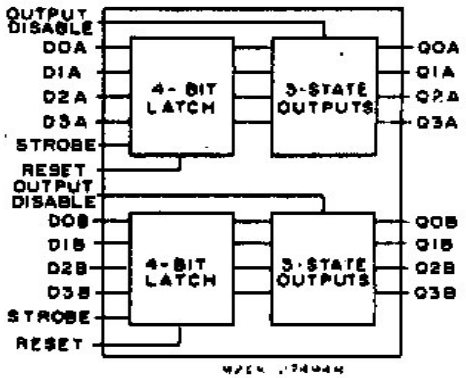
8 Bit Addressable Latch

CD4099B (File No. 948)



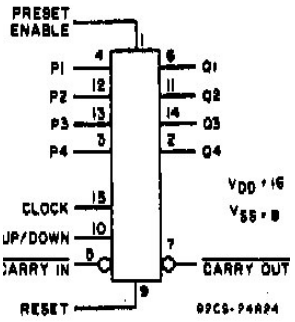
Strobed Hex Inverter/Buffer

CD4502B (File No. 1002)



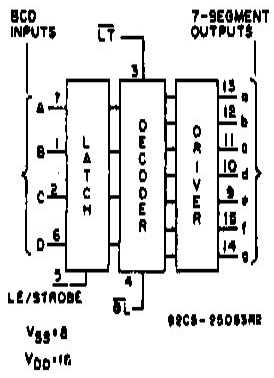
Dual 4-Bit Latch

CD4508B (File No. 1009)



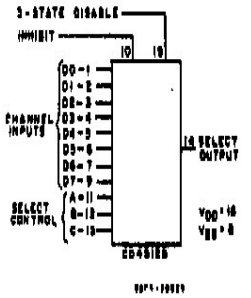
BCD Presettable Up/Down Counter

CD4510B (File No. 899)



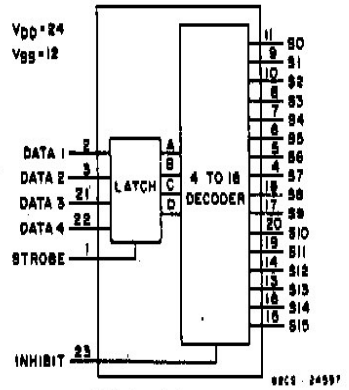
BCD-to-7-Segment Latch Decoder Driver

CD4511B (File No. 901)



8-Channel Data Selector

CD4512B (File No. 1032)

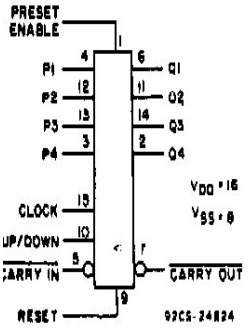


4-Bit Latch/4-to-16 Line Decoder

CD4514B (File No. 814) Output "High" on Select

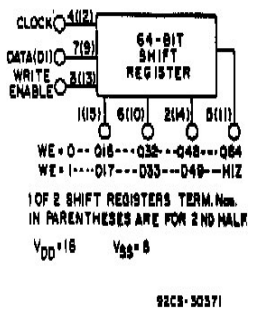
CD4515B (File No. 814) Output "Low" on Select

Functional Diagrams



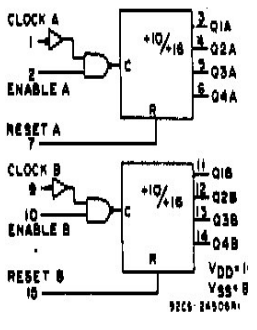
Binary Presettable Up/Down Counter

CD4516B (File No. 899)



Dual 64-Bit Shift Register

CD4517B* (Obj.)



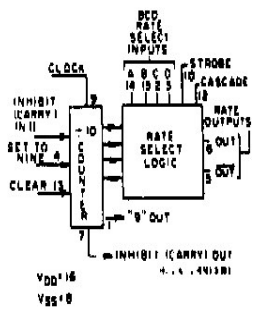
Dual Up Counter

CD4518B (File No. 808)

BCD

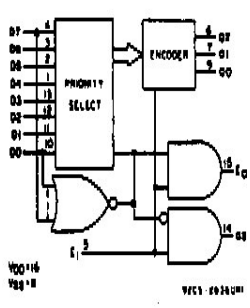
CD4520B (File No. 808)

Binary



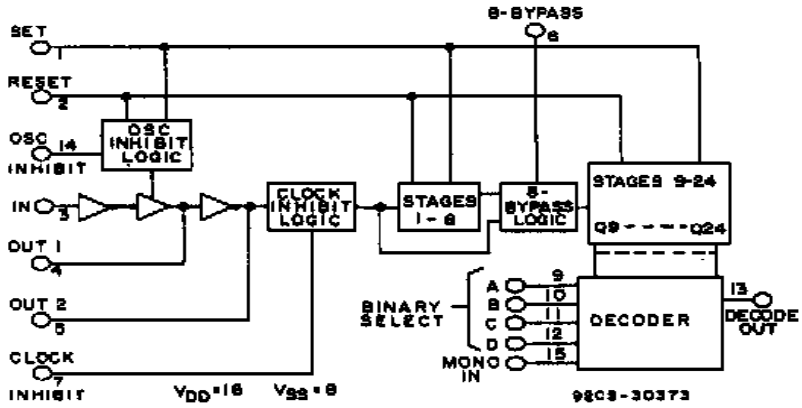
BCD Rate Multiplier

CD4527B (File No. 1006)



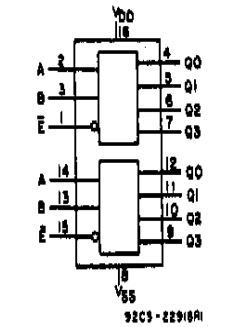
8-Bit Priority Encoder

CD4532B (File No. 876)



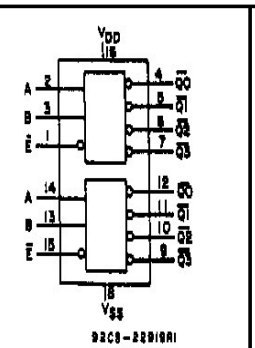
Programmable Timer

CD4536B* (Obj.)



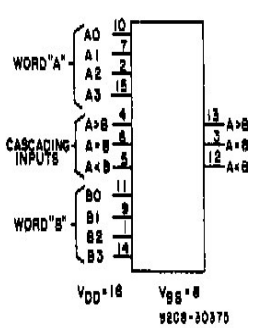
Dual Binary-to-1-of-4 Decoder/Demultiplexer Output "High" on Select

CD4555B (File No. 868)



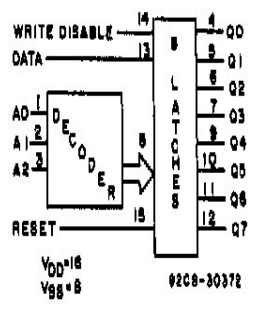
Dual Binary-to-1-of-4 Decoder/Demultiplexer Output "Low" on Select

CD4556B (File No. 868)



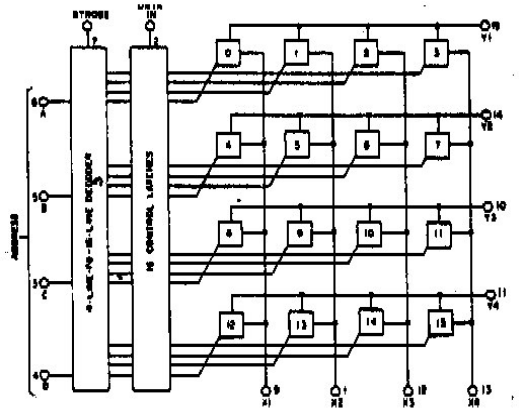
4-Bit Magnitude Comparator

CD4585B* (Obj.)



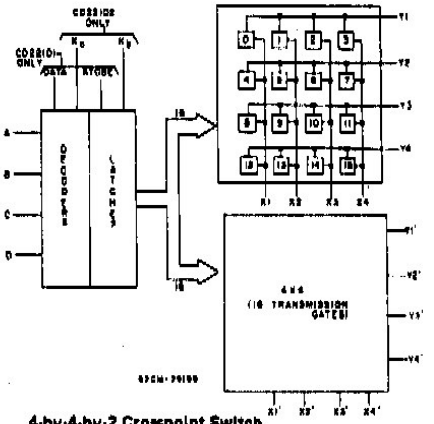
8-Bit Addressable Latch

Functional Diagrams



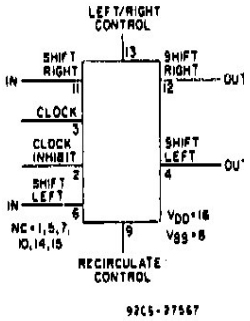
4-by-4 Crosspoint Switch with Control Memory

CD22100 (File No. 1076)



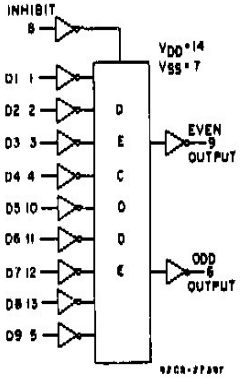
4-by-4-by-2 Crosspoint Switch with Control Memory

CD22101 (Prel.) CD22102 (Prel.)



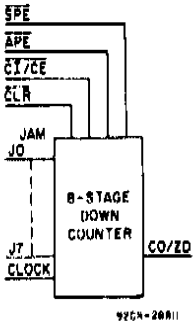
32-Stage Static Left/Right Shift Register

CD40100B (File No. 980)



9-Bit Parity Generator/Checker

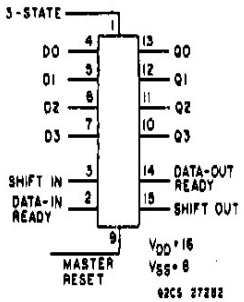
CD40101B (File No. 1000)



8-Stage Presettable Synchronous Down Counter

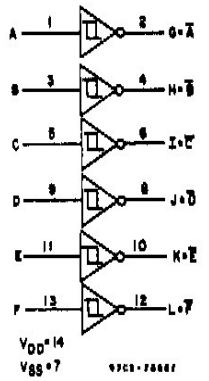
CD40102B (File No. 984) 2-Decade BCD

CD40103B (File No. 984) 8-Bit Binary



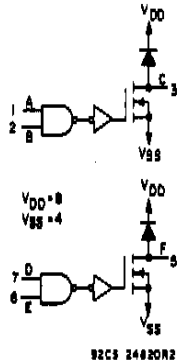
FIFO Register 4-Bits Wide by 16-Bits Long

CD40105B (File No. 1044



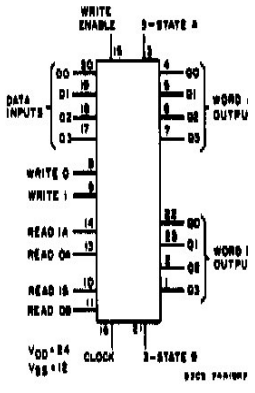
Hex Schmitt Trigger

CD40106B (File No. 1017)



Dual 2-Input NAND Buffer/Driver

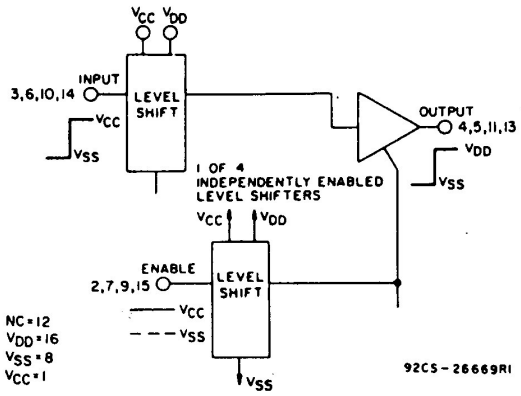
CD40107B (File No. 1015)



4-by-4 Multiport Register

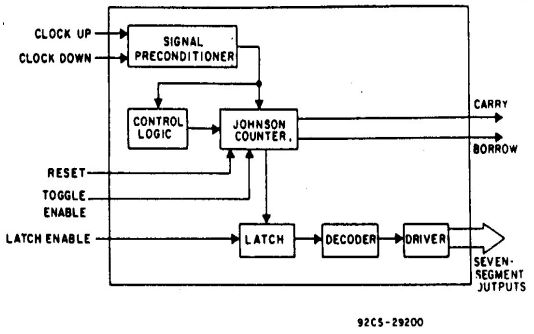
CD40108B (File No. 1011)

Functional Diagrams



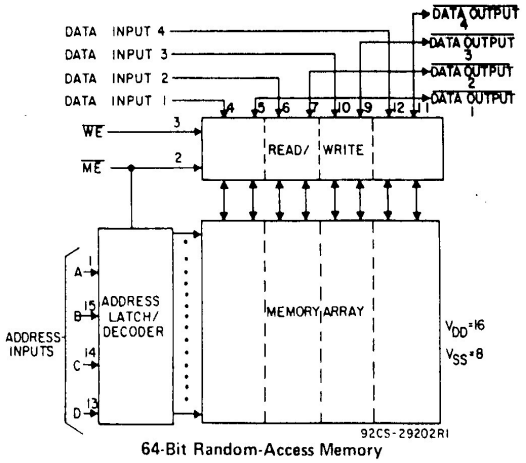
Quad Low-to-High Voltage Level Shifter

CD40109B (File No. 1018)



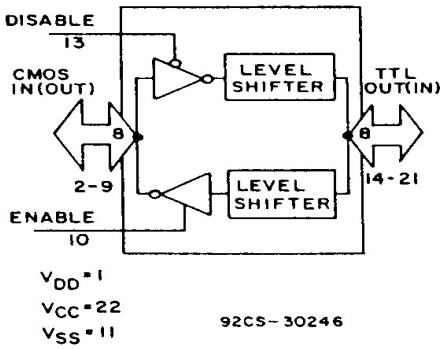
Decade Up-Down Counter/ Decoder/Latch/Driver

CD40110B (Obj.)



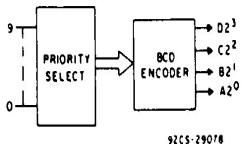
64-Bit Random-Access Memory

CD40114B■ (Obj.)



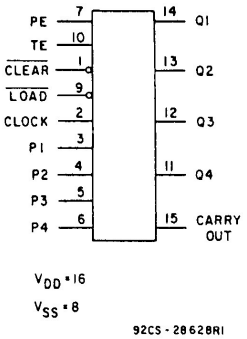
8-Bit Bidirectional CMOS/TTL Level Converter

CD40115 (File No. 1075)



10-Line-to-4-Line BCD Priority Encoder

CD40147B (Obj.)



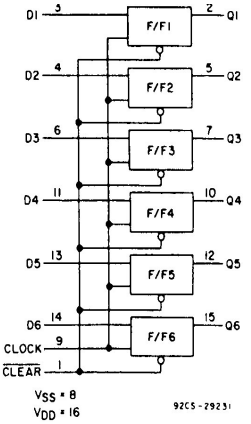
Synchronous 4-Bit Counter

CD40160B (File No. 1047) Decade with Asynchronous Clear

CD40161B (File No. 1047) Binary with Asynchronous Clear

CD40162B (File No. 1047) Decade with Synchronous Clear

CD40163B (File No. 1047) Binary with Synchronous Clear

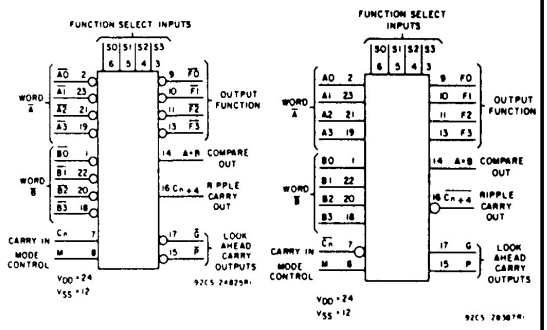


Hex "D" Type Flip-Flop

CD40174B (File No. 1031)

■ **To be announced**

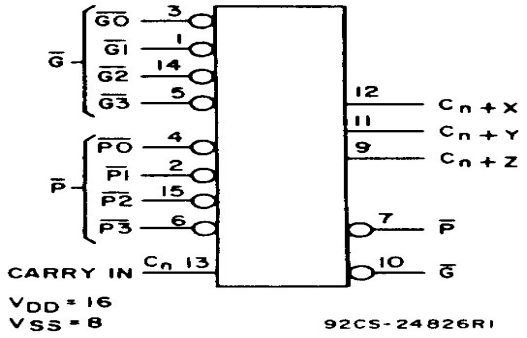
Functional Diagrams



4-Bit Arithmetic Logic Unit

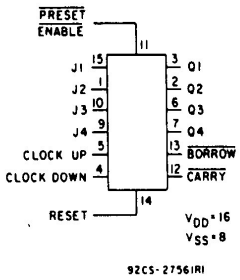
Active-Low Data | Active-High Data

CD40181B (File No. 989)



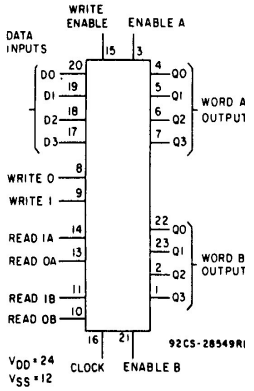
Look-Ahead Carry Generator

CD40182B (File No. 1008)



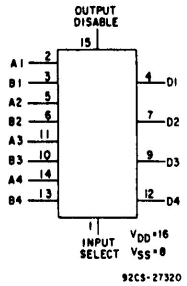
Presettable Up/Down Counter
(Dual Clock with Reset)

CD40192B (File No. 993) BCD | CD40193B (File No. 993) Binary



4-by-4 Multiport Register

CD40208B (File No. 1007)



Quad 2-Line-to-1-Line
Data Selector/Multiplexer

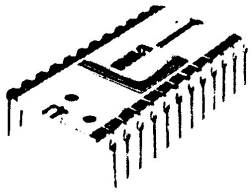
CD40257B (File No. 982)

11.3 Packages and Ordering Information

• Packages

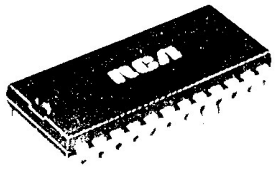
D Suffix
Ceramic Dual-In-Line Packages

Welded-Seal 14, 16, 24, and 28-lead versions



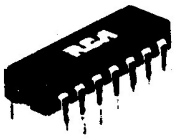
Side-Brazed 14, 16, 18, 22, and 24-lead versions

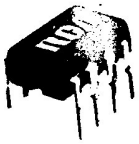
F Suffix
Frit-Seal Ceramic Dual-In-Line Packages



14, 16, and 24-lead versions

E Suffix
Plastic Dual-In-Line Packages





MiniDIP

8, 14, 16, 18, 22, and 24-lead versions

T Suffix
12-Lead TO-5 Style Package

CD4062A only

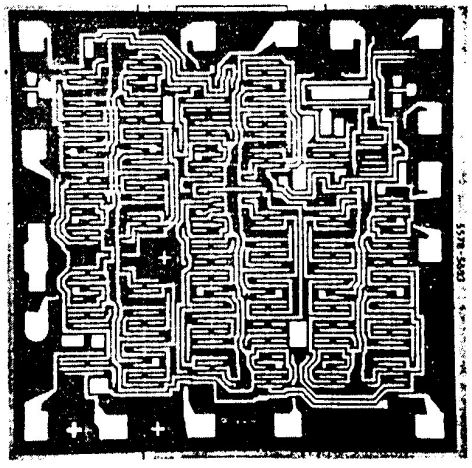
• Ordering Information

Most RCA COS/MOS integrated circuits are available in the following package styles and are identified by the Suffix Letters indicated below: dual-in-line ceramic, dual-in-line frit-seal ceramic, dual-in-line plastic, and in chip form. Some types are only available in one or two package styles. The available package styles for any specific type are given in the technical data for that type.

When ordering COS/MOS devices, it is important that the appropriate suffix letter be affixed to the type number of the device required. For example, a CD4016B in a dual-in-line ceramic package will be identified as the CD4016BD.

PACKAGE	SUFFIX LETTERS
Dual-In-Line Welded-Seal or Side-Brazed Ceramic	D
Dual-In-Line Frit-Seal Ceramic	F
Dual-In-Line plastic	E
TO-5 Style (CD4062A only)	T
Chip	H

H Suffix
COS/MOS Chip



11.4 Comparisan Table for Integrated Circuits (Foreign Product-Siemens Type)

Digital integrated circuits

SN 7400 N	FLH 101
SN 7401 N	FLH 201
SN 7401 NS1	FLH 201 S
SN 7401 NS3	FLH 201 T
SN 7402 N	FLH 191
SN 7402 NS1	FLH 191 S
SN 7403 N	FLH 291
SN 7403 NS1	FLH 291 S
SN 7403 NS3	FLH 291 T
SN 7404 N	FLH 211
SN 7405 N	FLH 271
SN 7405 NS1	FLH 271 S
SN 7405 NS3	FLH 271 T
SN 7406 N	FLH 481
SN 7407 N	FLH 491
SN 7408 N	FLH 381
SN 7409 N	FLH 391
SN 7409 NS1	FLH 391 T
SN 7410 N	FLH 111
SN 7412 N	FLH 501
SN 7413 N	FLH 351
SN 7416 N	FLH 481 T
SN 7417 N	FLH 491 T
SN 7420 N	FLH 121
SN 7422 N	FLH 611
SN 7423 N	FLH 511
SN 7425 N	FLH 521
SN 7426 N	FLH 291 U
SN 7427 N	FLH 621
SN 7428 N	FLH 661
SN 7430 N	FLH 131
SN 7432 N	FLH 631
SN 7437 N	FLH 531
SN 7438 N	FLH 541
SN 7440 N	FLH 141
SN 7442 N	FLH 281
SN 7443 N	FLH 361
SN 7444 N	FLH 371
SN 7445 N	FLL 111
SN 7446 N	FLL 121
SN 7446 AN	FLL 121 U
SN 7447 N	FLL 121 T
SN 7447 AN	FLL 121 V
SN 7448 N	FLH 551
SN 7450 N	FLH 151
SN 7451 N	FLH 161
SN 7453 N	FLH 171
SN 7454 N	FLH 181
SN 7460 N	FLY 101
SN 7470 N	FLJ 101
SN 7472 N	FLJ 111
SN 7473 N	FLJ 121
SN 7474 N	FLJ 141
SN 7475 N	FLJ 151
SN 7476 N	FLJ 131
SN 7480 N	FLH 221
SN 7481 AN	FLQ 111
SN 7482 N	FLH 231
SN 7483 N	FLH 241
SN 7484 AN	FLQ 121
SN 7485 N	FLH 431
SN 7486 N	FLH 341
SN 74H87 N	FLH 441
SN 7488 N	FLR 101
SN 7489 N	FLQ 101
SN 7490 N	FLJ 161
SN 7490 NS1	FLJ 161 S
SN 7491 AN	FLJ 221
SN 7492 N	FLJ 171
SN 7493 N	FLJ 181
SN 7494 N	FLJ 231
SN 7495 N	FLJ 191
SN 7496 N	FLJ 261
SN 7497 N	FLJ 331
SN 74100 N	FLJ 301
SN 74104 N	FLJ 281
SN 74105 N	FLJ 291
SN 74107 N	FLJ 271
SN 74110 N	FLJ 341
SN 74111 N	FLJ 351
SN 74115 N	FLJ 521
SN 74118 N	FLJ 361
SN 74119 N	FLJ 371
SN 74120 N	FLY 181
SN 74121 N	FLK 101
SN 74122 N	FLK 111
SN 74123 N	FLK 121
SN 74132 N	FLH 601
SN 74141 N	FLL 101
SN 74142 N	FLL 151
SN 74143 N	FLL 171
SN 74144 N	FLL 171 T
SN 74145 N	FLL 111 T
SN 74150 N	FLY 111
SN 74151 N	FLY 121
SN 74153 N	FLY 131
SN 74154 N	FLY 141
SN 74155 N	FLY 151
SN 74156 N	FLY 161
SN 74157 N	FLY 171
SN 74160 N	FLJ 401
SN 74161 N	FLJ 411
SN 74162 N	FLJ 421
SN 74163 N	FLJ 431
SN 74164 N	FLJ 441
SN 74165 N	FLJ 451
SN 74166 N	FLJ 461
SN 74167 N	FLJ 471
SN 74170 N	FLQ 131
SN 74174 N	FLJ 531
SN 74175 N	FLJ 541
SN 74180 N	FLH 421
SN 74181 N	FLH 401
SN 74182 N	FLH 411
SN 74H183 N	FLH 451
SN 74184 N	FLH 561
SN 74185 AN	FLH 571
SN 74187 N	FLR 111
SN 74188 N	(FLR 121)
SN 74190 N	FLJ 201
SN 74191 N	FLJ 211
SN 74192 N	FLJ 241
SN 74193 N	FLJ 251
SN 74194 N	FLJ 551
SN 74195 N	FLJ 561
SN 74196 N	FLJ 381
SN 74197 N	FLJ 391
SN 74198 N	FLJ 311
SN 74199 N	FLJ 321
SN 4929 N	FLH 251
SN 4930 N	FLH 321
SN 4931 N	FLH 331
SN 4932 N	FLJ 481
SN 4934 N	FLH 461
SN 4935 N	FLH 471
SN 49700 N	FLL 131
SN 49700 NS1	FLL 131 T
SN 49701 N	FLL 141
SN 49701 NS1	FLL 141 T
SN 49702 N	FLJ 491
SN 49703 N	FLH 641
SN 49704 N	FLJ 501
SN 49705 N	FLJ 511
SN 49713 N	FLH 731
SN 49713 NS1	FLH 731 T
DM 8200 N	FLH 311
DM 8220 N	FLH 651
JM 5600 CDE	FLR 121
JM 5523	FLQ 141

Analog integrated circuits

MC 1709 CG	TAA 521
MC 1709 CL	TAA 521 A
MC 1709 G	TAA 522
MC 1741 CG	TBA 221
MC 1741 G	TBA 222
OM 200	TAA 131
SN 5510 L	TAA 722
SN 7510 L	TAA 721
SN 72709 L	TAA 521
SN 72709 N	TAA 521 A
SN 52709 L	TAA 522
SN 72741 L	TBA 221
SN 72741 N	TBA 221 A
SN 72741 P	TBA 221 B
SN 52741 L	TBA 222
TAA 263	TAA 141
TAA 293	TAA 151
TOA 2709 V	TAA 521
TOA 2709 P	TAA 521 A
TOA 1709 V	TAA 522
TOA 2741 V	TBA 221
TOA 1741 V	TBA 222
µA 709	TAA 522
µA 709 C	TAA 521
µA 741	TBA 222
µA 741 C	TBA 221
SN 76660	TBA 120

Manufacturer:

SN ...	≙ Texas
AM ...	≙ NSC
JM ...	≙ Intersil
MC ...	≙ Motorola
TOA ...	≙ Transitron
µA ...	≙ Fairchild

The Series SN 8400 N corresponds to the Siemens-series FL. 105.
The desired type designations may be derived from the series SN 7400 N, e. g. SN 8401 = FLH 205, etc.

11.5 TTL Series (Transistor-Transistor-Logic)

General Information on the TTL-FL 100-7400 Series

➢ Static Data

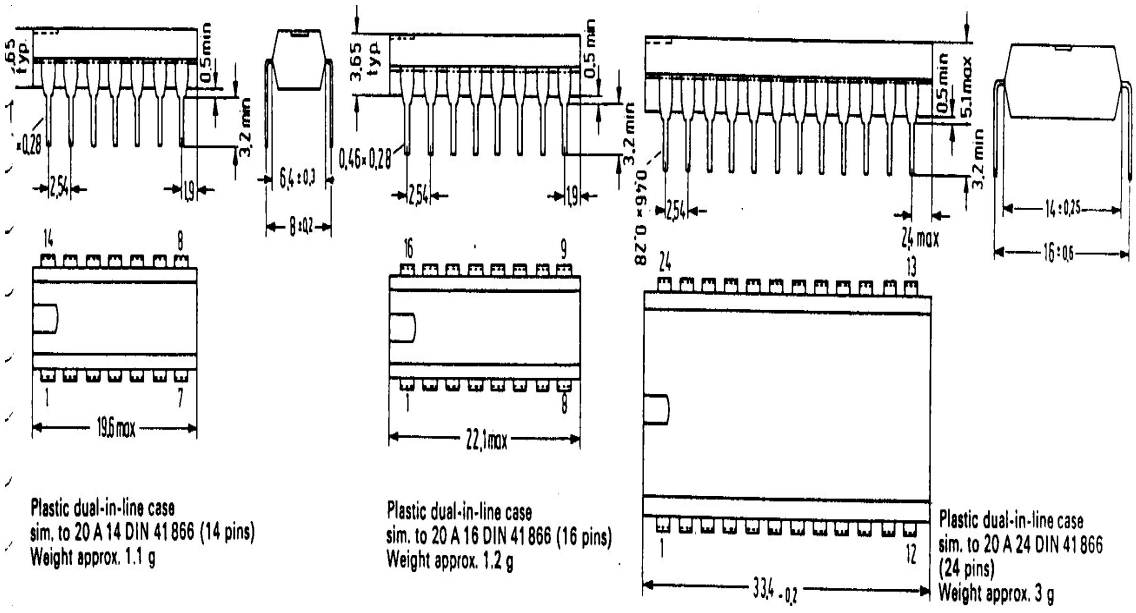
Maximum ratings are absolute values which, if exceded, can destroy the IC. Typical characteristics are statistically determined average values supplemented by a guaranteed apread (worst case). These values apply at a supply voltage V_S = 5V and an ambient temperature T_{amb} = 25°C, uniess otherwise stated.

The following maximum ratings apply to all types:		Min	Max	Unit.
Supply voltage	V_S	0	7	V
Input voltage	V_{in}	0	5,5	V
Operating temperature FL 101 (range 1)	T_{amb}	0	70	°C
Operating temperature FL 105 (range 5)	T_{amb}	- 25	85	°C
Storage temperature	Ts	- 65	150	°C
Thermal resistance (system – ambient air)	R_{thSamb}		150	K/W

➢ Dynamic Data

The switching times of the individual devices determine the maximum operating speed. Capacitive loading of the output and long lead lengths increase delay and thereby reduce the speed. Measurement of the propagation delay times t_{pd} is referred to threshold voltage V_{th} = 1.5 V, and is measured brtween the input and output. The rise time t_{TLH} and the fall time t_{THL} of the pulse is determined between the 10% and the and the 90% points.

Outline drawings of the TTL series



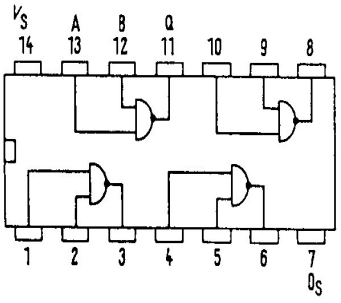
Plastic dual-in-line case
sim. to 20 A 14 DIN 41 866 (14 pins)
Weight approx. 1.1 g

Plastic dual-in-line case
sim. to 20 A 16 DIN 41 866 (16 pins)
Weight approx. 1.2 g

Plastic dual-in-line case
sim. to 20 A 24 DIN 41 866
(24 pins)
Weight approx. 3 g

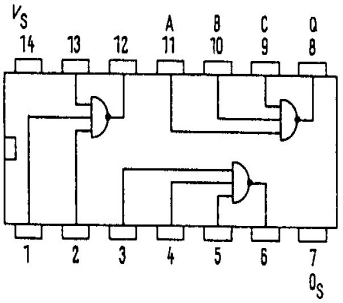
Pin configurations from top.

A. Gates



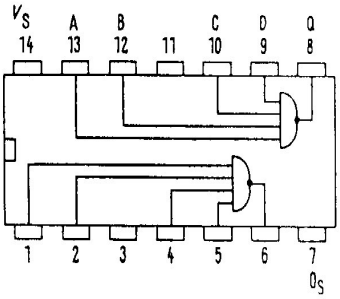
Quadruple 2-input NAND gate

FLH 101 - 7400 (Q67000 - H1)
FLH 105 - 8400 (Q67000 - H155)



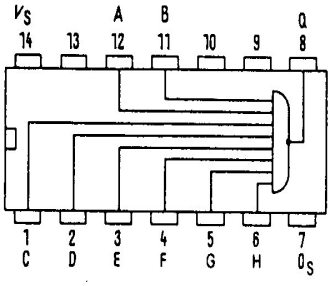
Triple 3-input NAND gate

FLH 111 - 7410 (Q67000 - H2)
FLH 115 - 8410 (Q67000 - H156)



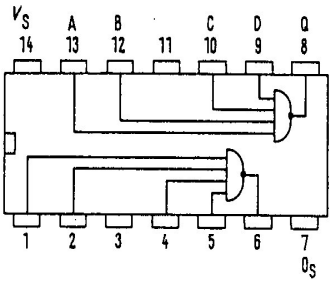
Dual 4-input NAND gate

FLH 121 - 7420 (Q67000 - H3)
FLH 125 - 8420 (Q67000 - H157)



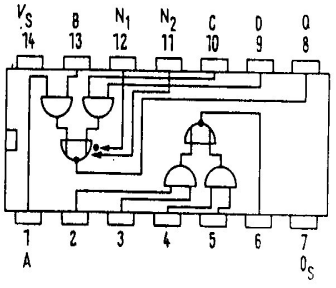
8-input NAND gate

FLH 131 - 7430 (Q67000 - H4)
FLH 135 - 8430 (Q67000 - H152)



Dual 4-input NAND power gate

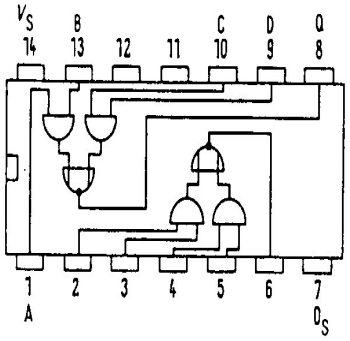
FLH 141 - 7440 (Q67000 - H5)
FLH 145 - 8440 (Q67000 - H158)



Expandable dual
2×2-input AND-OR-INVERT gate

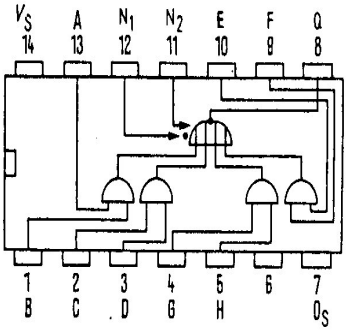
FLH 151 - 7450 (Q67000 - H6)
FLH 155 - 8450 (Q67000 - H159)

A. Gates



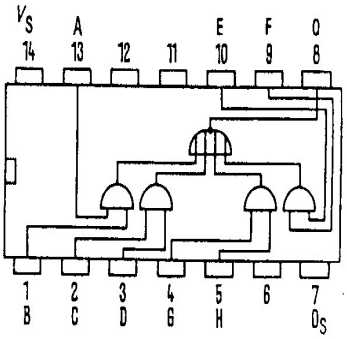
Dual 2×2-input
AND-OR-INVERT gate

FLH 161 - 7451 (Q67000 - H7)
FLH 165 - 8451 (Q67000 - H160)



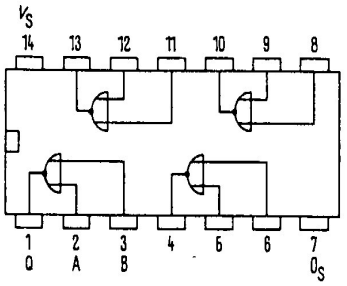
Expandable 4×2-input
AND-OR-INVERT gate

FLH 171 - 7453 (Q67000 - H8)
FLH 175 - 8453 (Q67000 - H161)



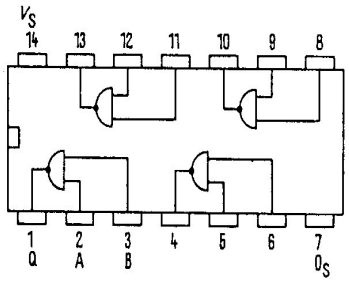
4×2-input AND-OR-INVERT gate

FLH 181 - 7454 (Q67000 - H9)
FLH 185 - 8454 (Q67000 - H162)



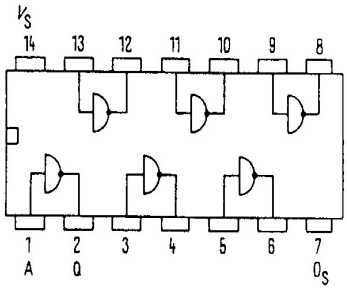
Quadruple 2-input NOR gate

FLH 191 - 7402 (Q67000 - H10)
FLH 191 S - 7402 S1 (Q67000 - H428)
FLH 195 - 8402 (Q67000 - H163)
FLH 195 S - 8402 S1 (Q67000 - H435)



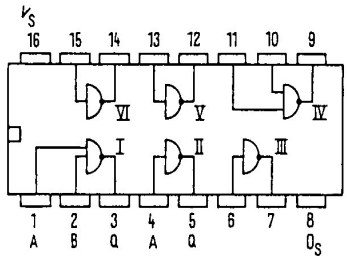
Quadruple 2-input NOR gate
with open collector output

FLH 201 - 7401 (Q67000 - H11)
FLH 201 S - 7401 S1 (Q67000 - H433)
FLH 201 T - 7401 S3 (Q67000 - H409)
FLH 205 - 8401 (Q67000 - H164)
FLH 205 S - 8401 S1 (Q67000 - H450)
FLH 205 T - 8401 S3 (Q67000 - H434)



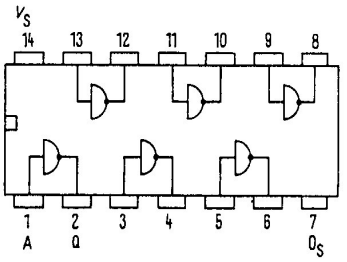
Hex inverter

FLH 211 - 7404 (Q67000 - H153)
FLH 215 - 8404 (Q67000 - H243)



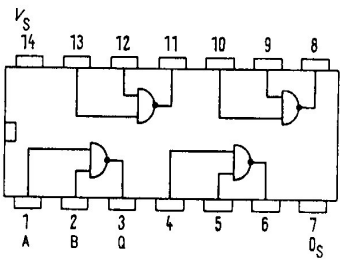
Quadruple inverter and
dual 2-input NAND gate

FLH 251 - 4929 (Q67000 - H138)
FLH 255 - 49829 (Q67000 - H247)



Hex inverter with
open collector output

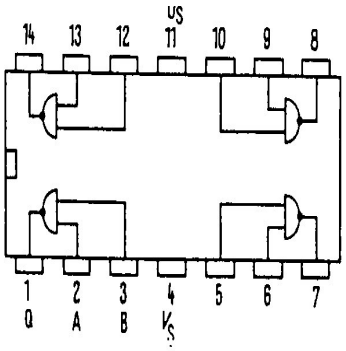
FLH 271 - 7405 (Q67000 - H154)
FLH 271 S - 7405 S1 (Q67000 - H462)
FLH 271 T - 7405 S3 (Q67000 - H459)
FLH 275 - 8405 (Q67000 - H248)
FLH 275 S - 8405 S1 (Q67000 - H460)
FLH 275 T - 8405 S3 (Q67000 - H461)



Quadruple 2-input NAND gate
with open collector output

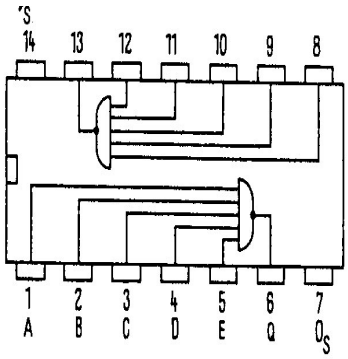
FLH 291 - 7403 (Q67000 - H149)
FLH 291 S - 7403 S1 (Q67000 - H242)
FLH 291 T - 7403 S3 (Q67000 - H429)
FLH 291 U - 7426 (Q67000 - H506)
FLH 295 - 8403 (Q67000 - H249)
FLH 295 S - 8403 S1 (Q67000 - H437)
FLH 295 T - 8403 S3 (Q67000 - H438)
FLH 295 U - 8426 (Q67000 - H507)

A. Gates



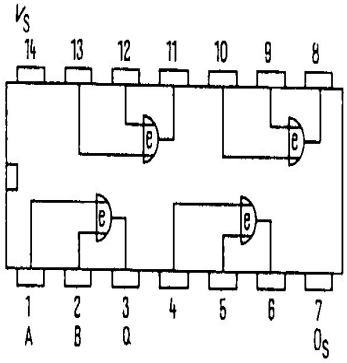
Quadruple 2-input NAND power gate

FLH 321 - 4930 (Q67000 - H139)
FLH 325 - 49830 (Q67000 - H315)



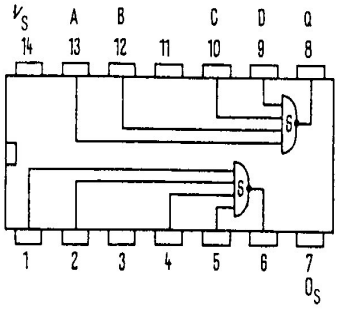
Dual 5-input NAND gate

FLH 331 - 4931 (Q67000 - H410)
FLH 335 - 49831 (Q67000 - H316)



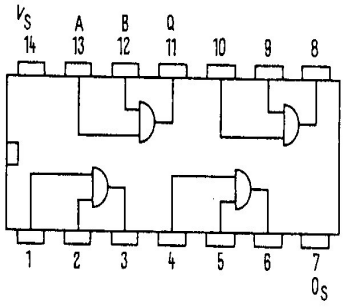
Quadruple 2-input
Exclusive-OR gate

FLH 341 - 7486 (Q67000 - H354)
FLH 345 - 8486 (Q67000 - H297)



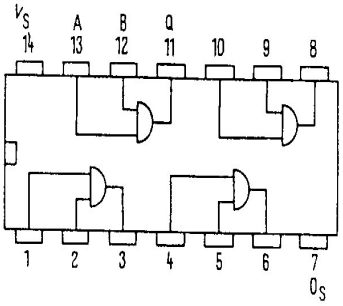
Dual 4-input NAND Schmitt trigger

FLH 351 - 7413 (Q67000 - L19)
FLH 355 - 8413 (Q67000 - H325)



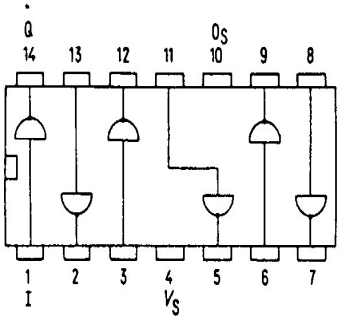
Quadruple 2-input AND gate

FLH 381 - 7408 (Q67000 - H398)
FLH 385 - 8408 (Q67000 - H402)



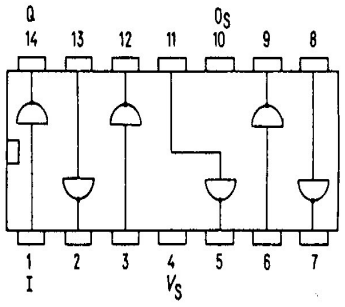
Quadruple 2-input AND gate
with open collector output

FLH 391 - 7409 (Q67000 - H399)
FLH 395 - 8409 (Q67000 - H425)
FLH 391 T - 7409 S1 (Q67000 - H720)
FLH 395 T - 8409 S1 (Q67000 - H846)



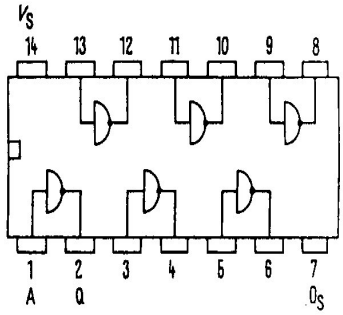
Hex inverter with expander node
and open collector output

FLH 461 - 4934 (Q67000 - H141)
FLH 465 - 49834 (Q67000 - H142)



Hex inverter with expander node

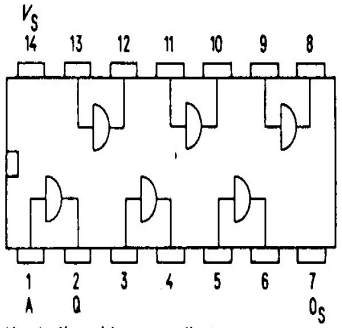
FLH 471 - 4935 (Q67000 - H512)
FLH 475 - 49835 (Q67000 - H513)



Hex buffer, inverting with open
collector output for 40 mA

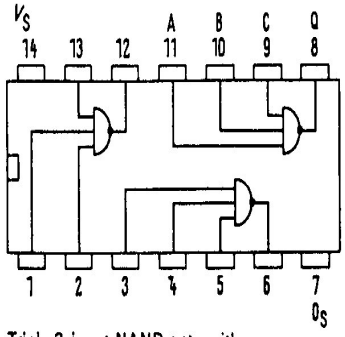
FLH 481 - 7406 (Q67000 - H396)
FLH 485 - 8406 (Q67000 - H400)
FLH 481 T - 7416 (Q67000 - H440)
FLH 485 T - 8416 (Q67000 - H443)

A. Gates



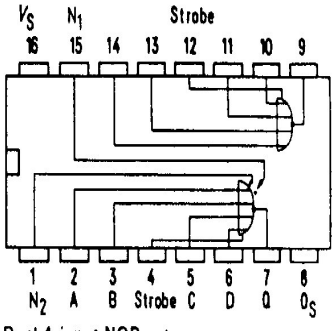
Hex buffer with open collector output for 40 mA

FLH 491 - 7407 (Q67000 - H397)
FLH 495 - 8407 (Q67000 - H401)
FLH 491 T - 7417 (Q67000 - H441)
FLH 495 T - 8417 (Q67000 - H444)



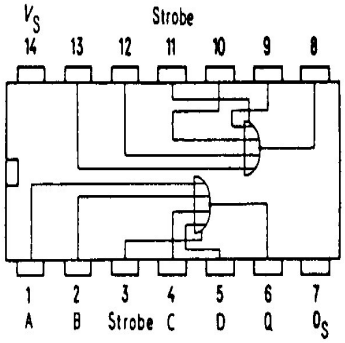
Triple 3-input NAND gate with open collector output

FLH 501 - 7412 (Q67000 - H526)
FLH 505 - 8412 (Q67000 - H527)



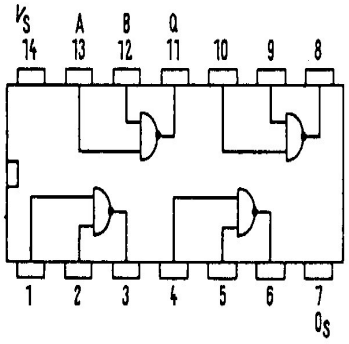
Dual 4-input NOR gate with strobe and expander node

FLH 511 - 7423 (Q67000 - H497)
FLH 515 - 8423 (Q67000 - H518)



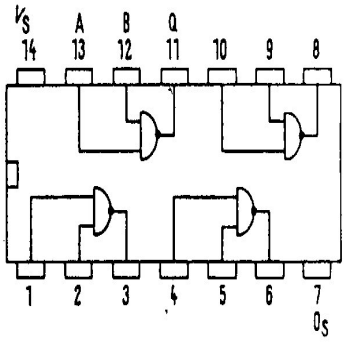
Dual 4-input NOR gate with strobe

FLH 521 - 7425 (Q67000 - H489)
FLH 525 - 8425 (Q67000 - H519)



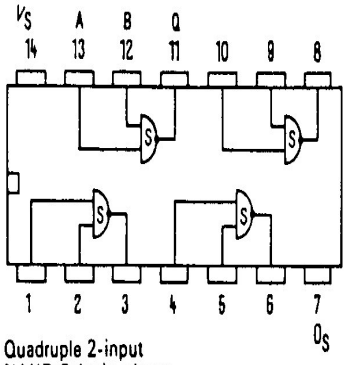
Quadruple 2-input NAND power gate

FLH 531 - 7437 (Q67000 - H416)
FLH 535 - 8437 (Q67000 - H520)



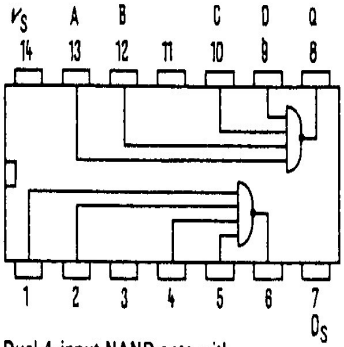
Quadruple 2-input NAND power gate with open collector output

FLH 541 - 7438 (Q67000 - H493)
FLH 545 - 8438 (Q67000 - H521)



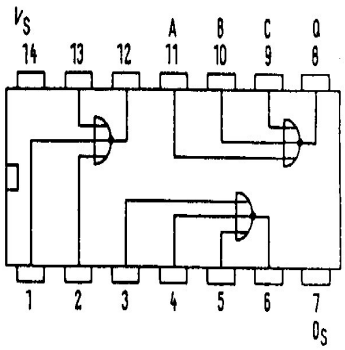
Quadruple 2-input NAND Schmitt trigger

FLH 601 - 74132 (Q67000 - H618)
FLH 605 - 84132 (Q67000 - H624)



Dual 4-input NAND gate with open collector output

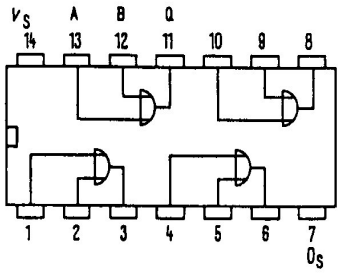
FLH 611 - 7422 (Q67000 - H625)
FLH 615 - 8422 (Q67000 - H626)



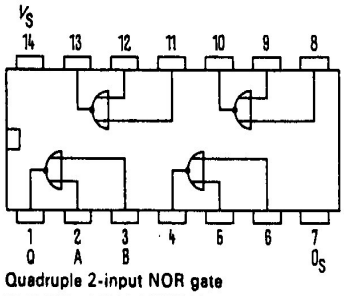
Triple 3-input NOR gate

FLH 621 - 7427 (Q67000 - H586)
FLH 625 - 8427 (Q67000 - H589)

A. Gates

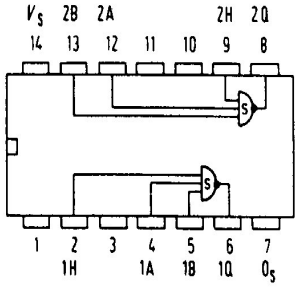


Quadruple 2-input OR gate
FLH 631 - 7432 (Q67000 - H491)
FLH 635 - 8432 (Q67000 - H593)

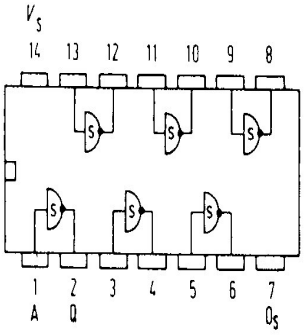


Quadruple 2-input NOR gate
FLH 661 - 7428 (Q67000 - H490)
FLH 665 - 8428 (Q67000 - H590)

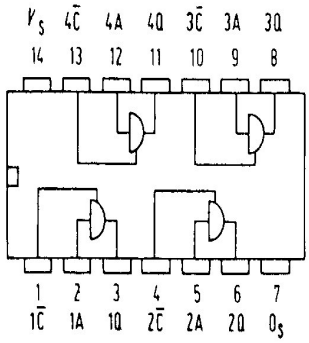
Quadruple 2-input NOR gate with open collector output
7433 (Q67000-H492)
8433 (Q67000-H591)
74128 (Q67000-H723)
84128 (Q67000-H853)



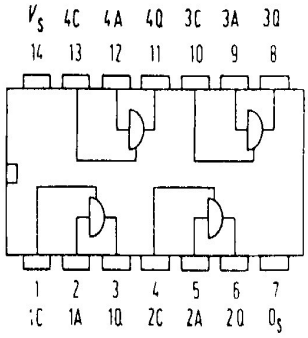
Dual NAND Schmitt trigger with high input impedance
FLH 731 - 49713 (Q67000 - H678)
FLH 735 - 49813 (Q67000 - H843)
FLH 731 T - 49713 S1 (Q67000 - H726)
FLH 735 T - 49813 S1 (Q67000 - H844)



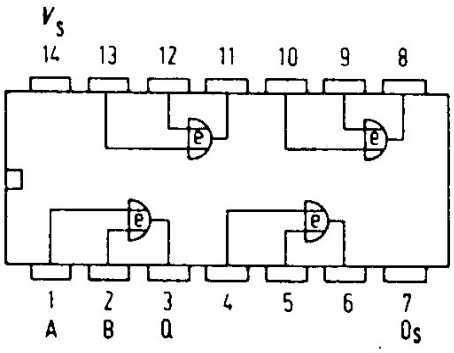
Hex NAND Schmitt trigger
7414 (Q67000-L109)
8414 (Q67000 - L120)



Quadruple AND gate with tri-state outputs
74125 (Q67000 - H721)
84125 (Q67000 - H851)

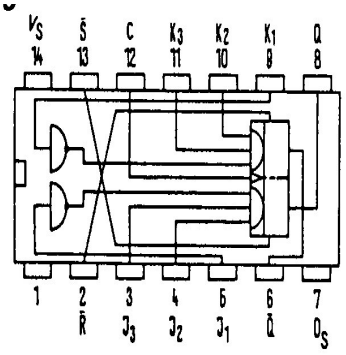


Quadruple AND gate with tri-state outputs
74126 (Q67000 - H722)
84126 (Q67000 - H852)



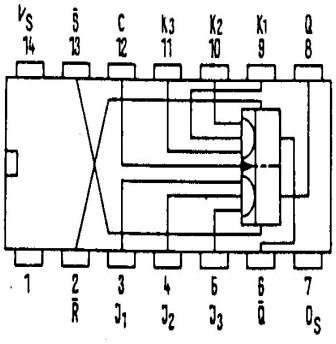
Quadruple 2-input exclusive-OR gate with open collector output
74136 (Q67000 - J512)
84136 (Q67000 - J516)

B. Flipflops and Memories



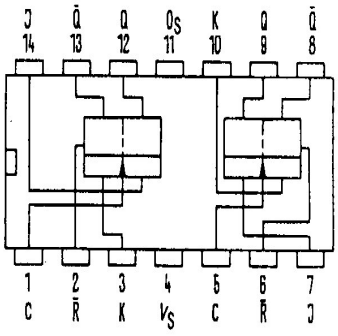
3+3-input JK-flipflop

FLJ 101 - 7470 (Q67000 - J1)
FLJ 105 - 8470 (Q67000 - J80)



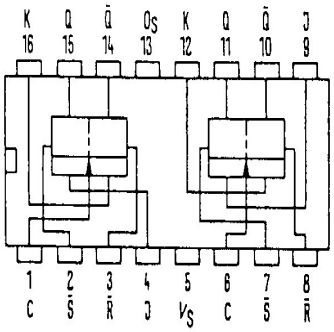
3+3-input JK-master-slave-flipflop

FLJ 111 - 7472 (Q67000 - J2)
FLJ 115 - 8472 (Q67000 - J81)



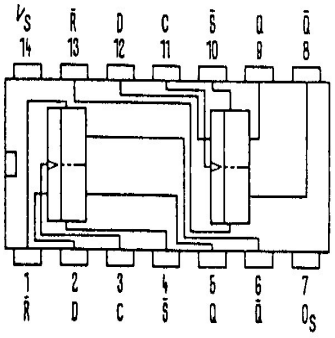
Dual JK-master-slave-flipflop
with reset

FLJ 121 - 7473 (Q67000 - J3)
FLJ 125 - 8473 (Q67000 - J79)



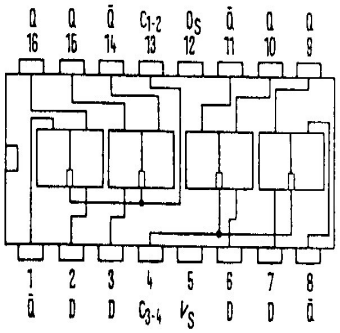
Dual JK-master-slave-flipflop
with set and reset

FLJ 131 - 7476 (Q67000 - J4)
FLJ 135 - 8476 (Q67000 - J82)



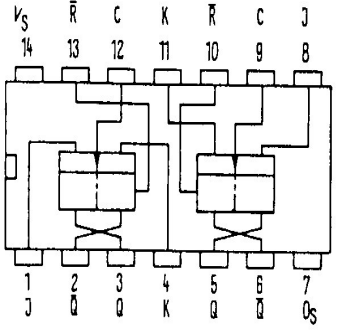
Dual D-flipflop

FLJ 141 - 7474 (Q67000 - J9)
FLJ 145 - 8474 (Q67000 - J83)



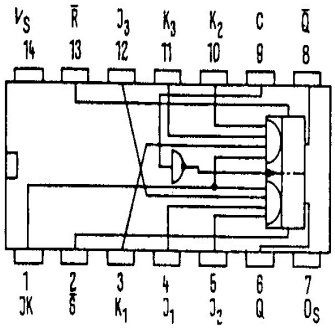
Quadruple D-flipflop

FLJ 151 - 7475 (Q67000 - J5)
FLJ 155 - 8475 (Q67000 - J84)



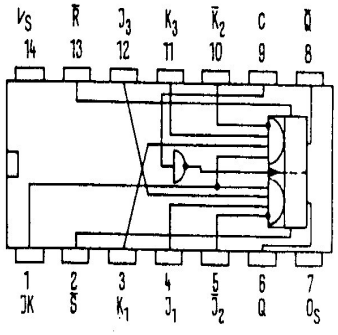
Dual JK-master-slave-flipflop
with reset

FLJ 271 - 74107 (Q67000 - J165)
FLJ 275 - 84107 (Q67000 - J222)



JK-master-slave-flipflop
with JK input

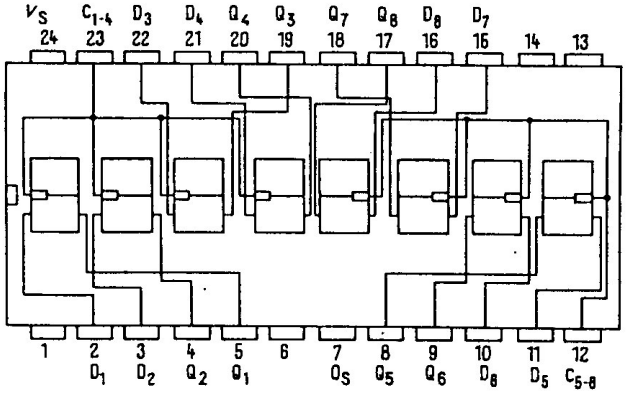
FLJ 281 - 74104 (Q67000 - J241)



JK-master-slave-flipflop
with J, K, and JK-inputs

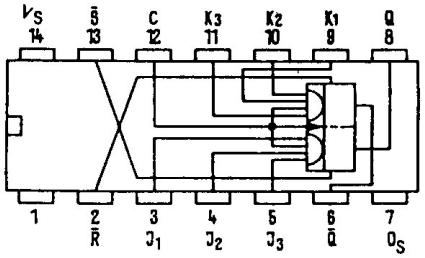
FLJ 291 - 74105 (Q67000 - J242)

B. Flipflops and Memories



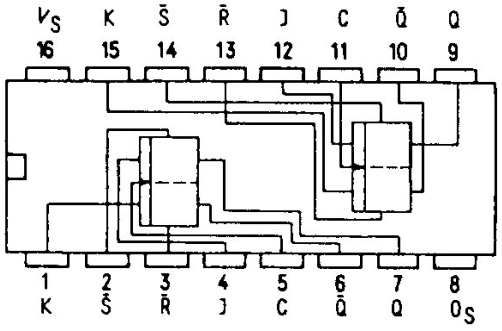
Eight D-flipflop

FLJ 301 - 74100 (Q67000 - J164)
FLJ 305 - 84100 (Q67000 - J233)



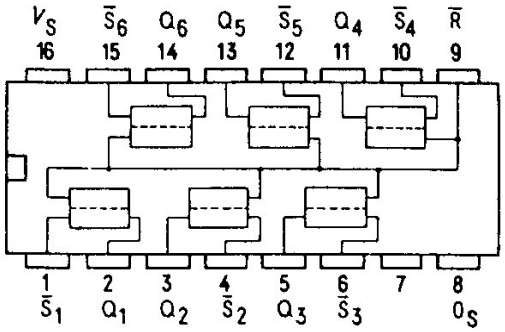
JK-master-slave-flipflop
with data lockout

FLJ 341 - 74110 (Q67000 - J224)
FLJ 345 - 84110 (Q67000 - J274)



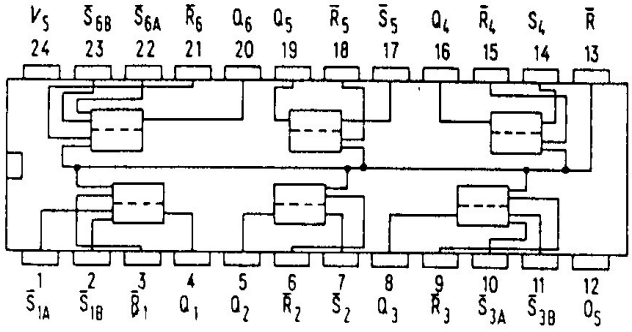
Dual JK-master-slave-flipflop
with data lockout.
set and reset

FLJ 351 - 74111 (Q76000 - J225)
FLJ 355 - 84111 (Q67000 - J289)



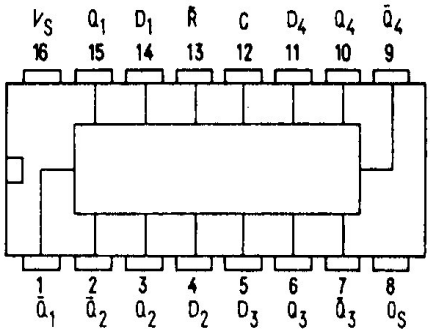
Hex RS-flipflop
with common reset

FLJ 361 - 74118 (Q67000 - J275)
FLJ 365 - 84118 (Q67000 - J290)



Hex RS-flipflop
with set and reset

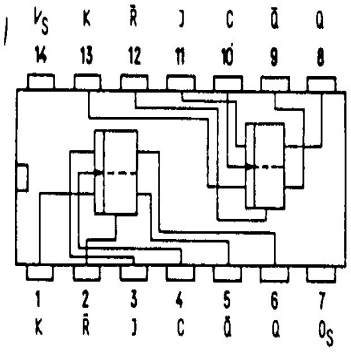
FLJ 371 - 74119 (Q67000 - J276)
FLJ 375 - 84119 (Q67000 - J291)



Quadruple D-flipflop
with common reset

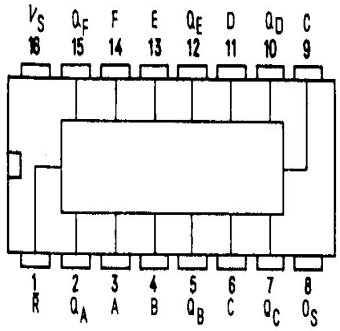
FLJ 491 - 49702 (Q67000 - H259)
FLJ 495 - 4982 (Q67000 - J151)

B. Flipflops and Memories



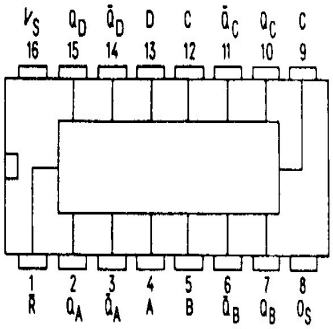
Dual JK-master-slave-flipflop
with reset

FLJ 521 - 74115 (Q67000 - J306)
FLJ 525 - 84115 (Q67000 - J376)



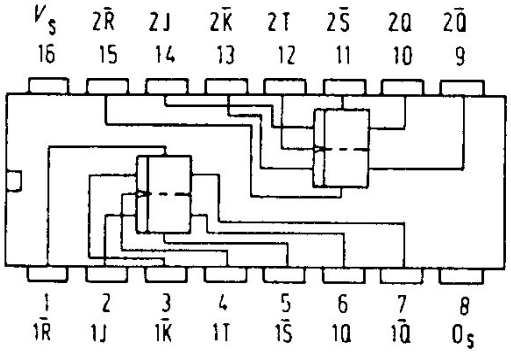
Hex D-flipflop with reset

FLJ 531 - 74174 (Q67000 - J349)
FLJ 535 - 84174 (Q67000 - J378)



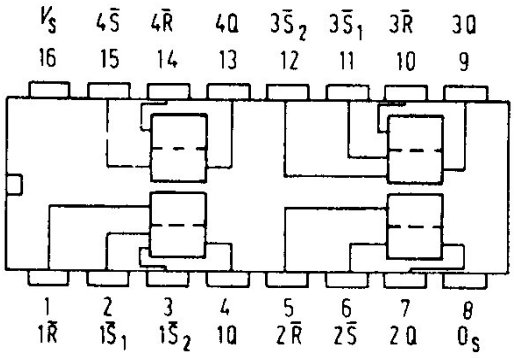
Quadruple D-flipflop with reset

FLJ 541 - 74175 (Q67000 - J379)
FLJ 545 - 84175 (Q67000 - J380)



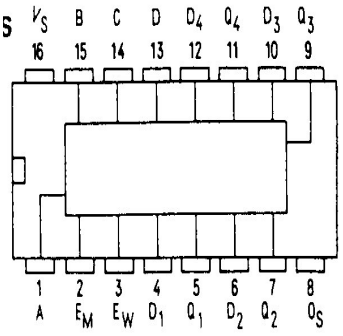
Dual JK-flipflop
with reset and set

74109 (Q67000 - J431)
84109 (Q67000 - J514)



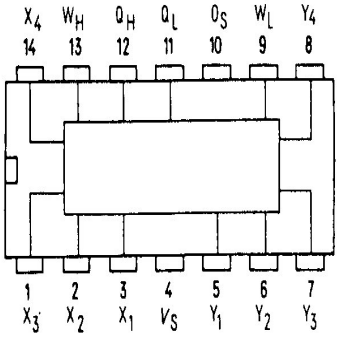
Quadruple RS-flipflop
with separate reset

74279 (Q67000 - J580)



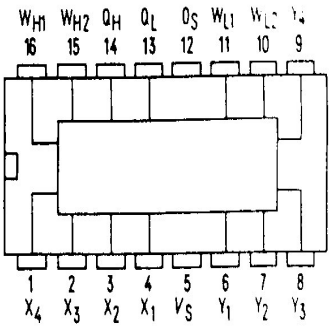
64-Bit RAM (Random Access Memory)
with open collector outputs

FLQ 101 - 7489 (Q67000 - Q11)
FLQ 105 - 8489 (Q67000 - Q12)



16-Bit RAM

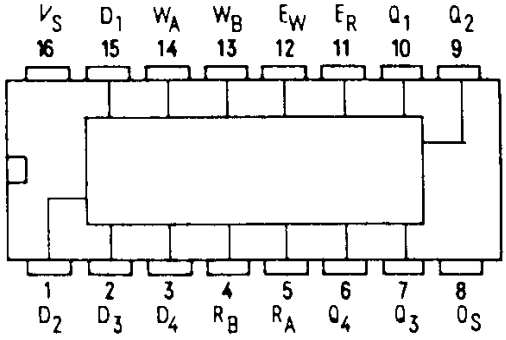
FLQ 111 - 7481 A (Q67000 - J27)
FLQ 115 - 8481 A (Q67000 - J250)



16-Bit RAM with 2 read
and write inputs each

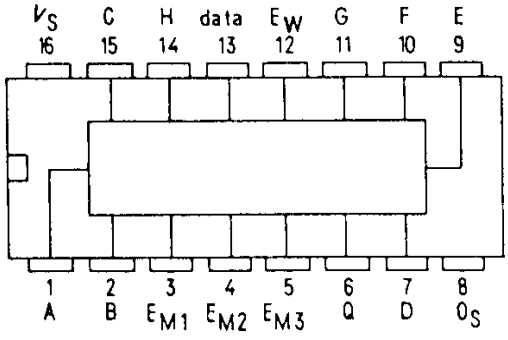
FLQ 121 - 7484 A (Q67000 - J163)
FLQ 125 - 8484 A (Q67000 - Q5)

B. Flipflops and Memories



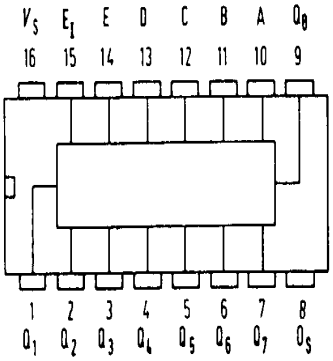
16-Bit RAM, 4 words of 4 bits

FLQ 131 - 74170 (Q67000 - Q6)
FLQ 135 - 84170 (Q67000 - Q25)



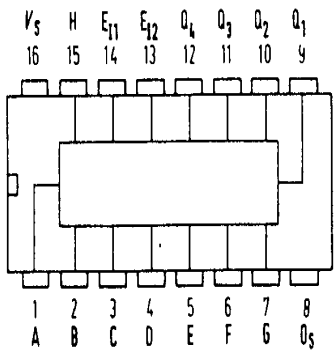
256-Bit RAM with tri-state output

FLQ 141 (Q67000 - Q20)



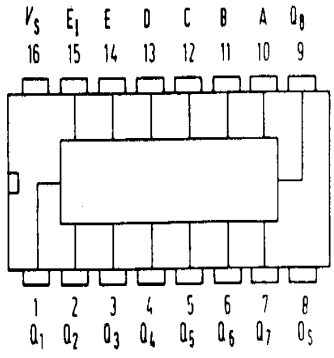
256-Bit ROM (Read-Only-Memory)

FLR 101 - 7488A (Q67000 - R1)
FLR 105 - 8488A (Q67000 - R2)



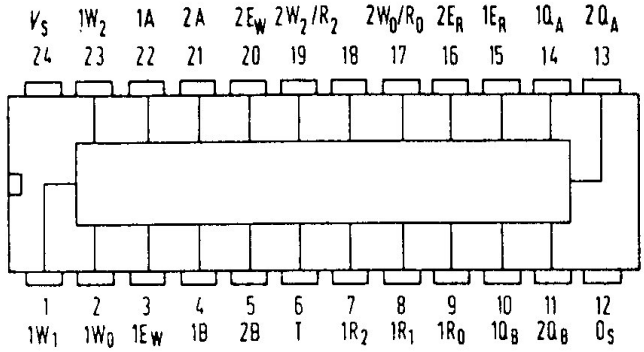
1024-Bit ROM

FLR 111 - 74187 (Q67000 - R5)



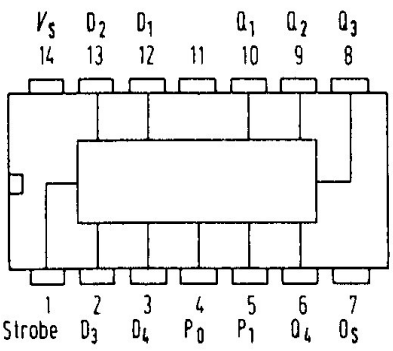
256-Bit ROM, programmable

FLR 121 - 74188 (Q67000 - R13)



16-Bit RAM with multiple input and output and tri-state outputs

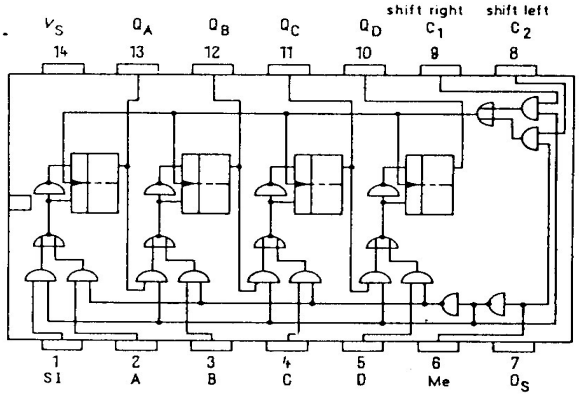
74172 (Q67000 - J432)



4-Bit priority register

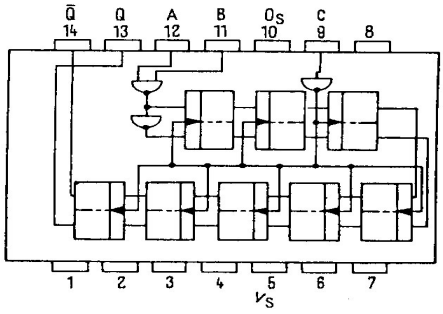
74278 (Q67000 - J576)

C. Shit-registers



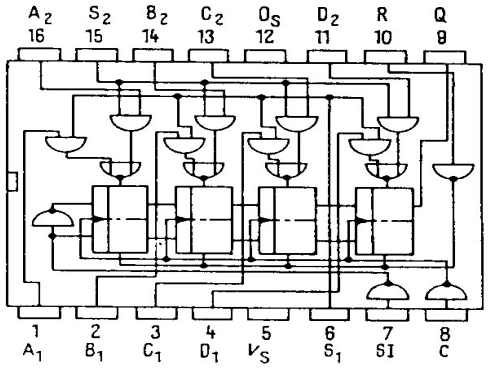
4-Bit shiftregister, reversible, parallel in/out

FLJ 191 - 7495A (Q67000 - Q427)
FLJ 195 - 8495A (Q67000 - Q513)



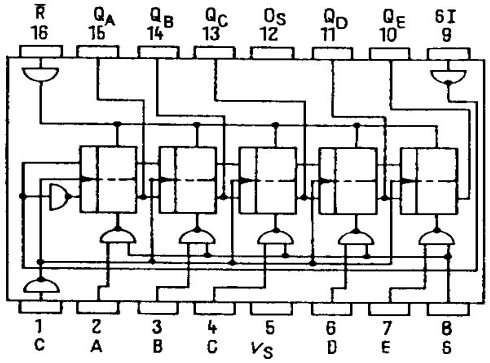
8-Bit shiftregister, serial in/out

FLJ 221 - 7491 A (Q67000 - J32)
FLJ 225 - 8491 A (Q67000 - J252)



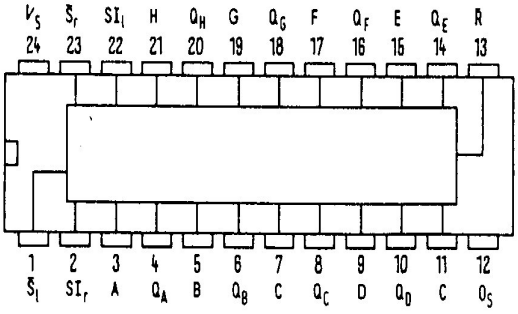
4-Bit shiftregister parallel in, serial out

FLJ 231 - 7494 (Q67000 - J35)
FLJ 235 - 8494 (Q67000 - J200)



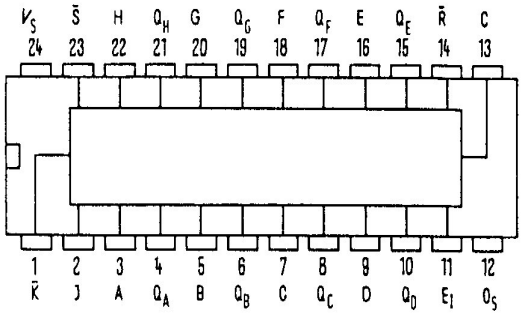
5-Bit shiftregister, parallel in; parallel out

FLJ 261 - 7496 (Q67000 - J37)
FLJ 265 - 8496 (Q67000 - J203)



Synchronous bidirectional 8-bit shiftregister, parallel in/out

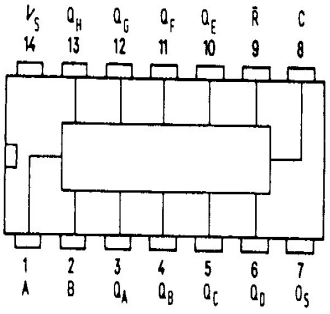
FLJ 311 - 74198 (Q67000 - J244)
FLJ 315 - 84198 (Q67000 - J235)



Synchronous 8-bit shiftregister with J-K-inputs, parallel in/out

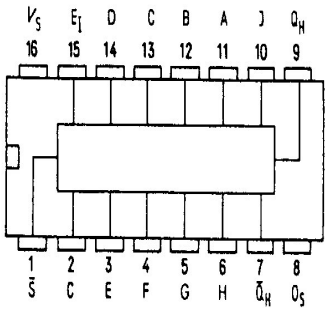
FLJ 321 - 74199 (Q67000 - J245)
FLJ 325 - 84199 (Q67000 - J237)

C. Shit-registers



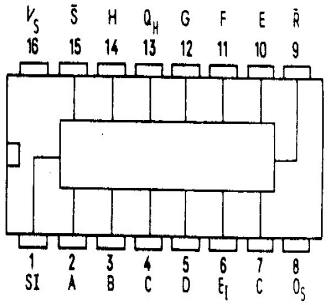
8-Bit shiftregister, parallel out

FLJ 441 - 74164 (Q67000 - J280)
FLJ 445 - 84164 (Q67000 - J298)



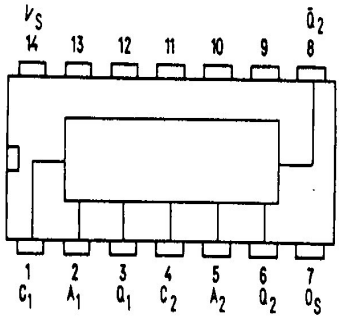
8-Bit shiftregister, parallel in

FLJ 451 - 74165 (Q67000 - J281)
FLJ 455 - 84165 (Q67000 - J299)



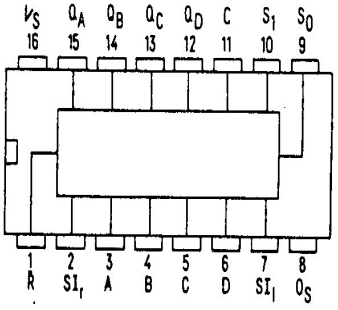
Synchronous 8-bit shiftregister, parallel in, serial out

FLJ 461 - 74166 (Q67000 - J282)
FLJ 465 - 84166 (Q67000 - J300)



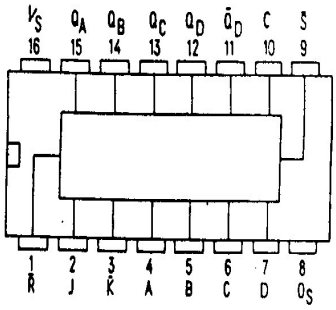
Dual 8-bit shiftregister, serial in/out

FLJ 481 - 4932 (Q67000 - J72)
FLJ 485 - 49832 (Q67000 - J152)



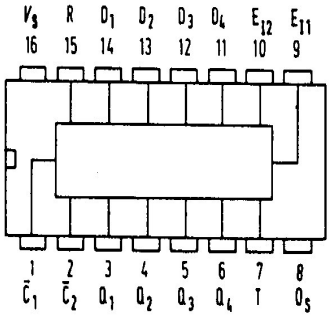
Synchronous 4-bit shiftregister, parallel in/out, right/left

FLJ 551 - 74194 (Q67000 - J350)
FLJ 555 - 84194 (Q67000 - J382)



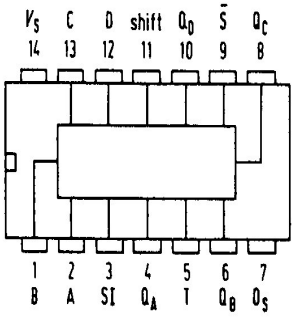
Synchronous 4-bit shiftregister, parallel in/out, with J-K inputs

FLJ 561 - 74195 (Q67000 - J286)
FLJ 565 - 84195 (Q67000 - J384)



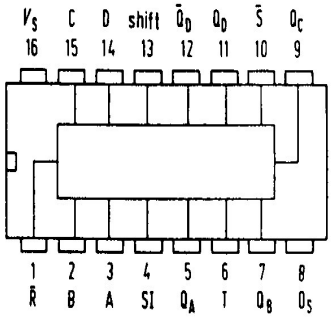
4-Bit shiftregister with tri-state outputs

74173 (Q67000 - J433)
84173 (Q67000 - J518)



4-Bit shiftregister, parallel in/out

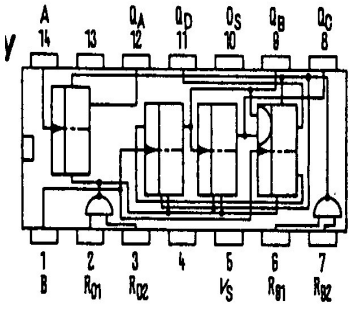
74178 (Q67000 - J436)
84178 (Q67000 - J519)



4-Bit shiftregister, parallel in/out

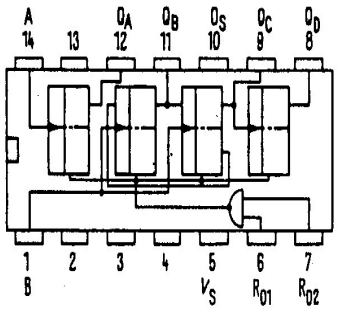
74179 (Q67000 - J437)
84179 (Q67000 - J520)

D. Counters and Frequency Dividers



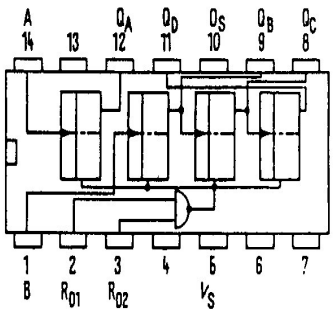
Decimal counter

FLJ 161 - 7490 (Q67000 - J10)
FLJ 161 S - 7490 S1 (Q67000 - J248)
FLJ 165 - 8490 (Q67000 - J85)
FLJ 165 S - 8490 S1 (Q67000 - J249)



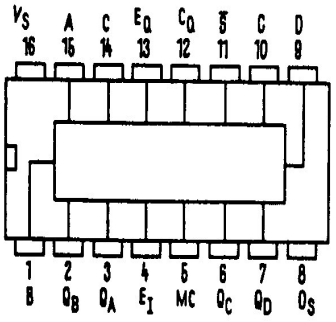
Divide-by-twelve counter

FLJ 171 - 7492 (Q67000 - J33)
FLJ 175 - 8492 (Q67000 - J122)



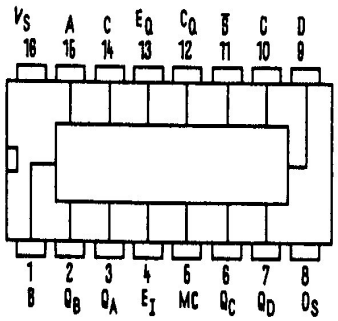
Binary counter

FLJ 181 - 7493 (Q67000 - J34)
FLJ 185 - 8493 (Q67000 - J123)



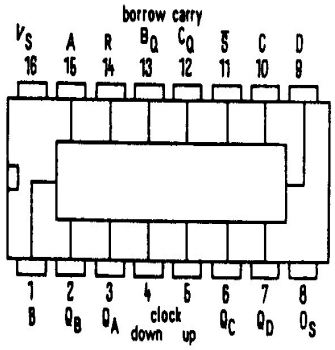
Decimal counter, reversible with set and reset

FLJ 201 - 74190 (Q67000 - J144)
FLJ 205 - 84190 (Q67000 - J198)



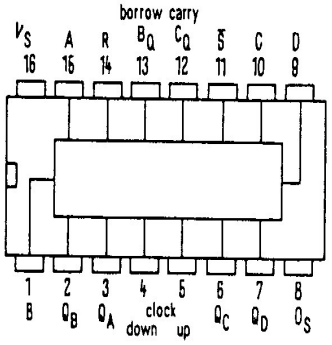
Binary counter, reversible with set and reset

FLJ 211 - 74191 (Q67000 - J145)
FLJ 215 - 84191 (Q67000 - J199)



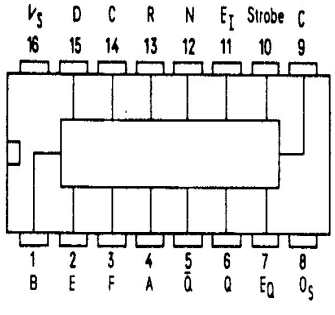
Decimal counter, reversible with clock inputs for up and down count

FLJ 241 - 74192 (Q67000 - J174)
FLJ 245 - 84192 (Q67000 - J201)



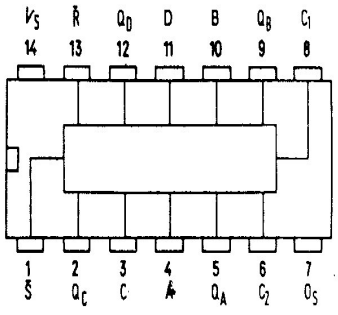
Binary counter, reversible with clock inputs for up and down count

FLJ 251 - 74193 (Q67000 - J175)
FLJ 255 - 84193 (Q67000 - J202)



Programmable 6-bit rate multiplier

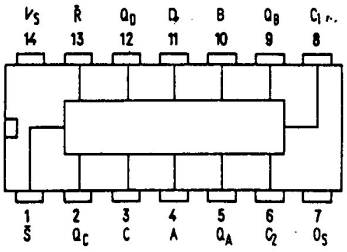
FLJ 331 - 7497 (Q67000 - J240)



Decimal counter with set and reset for 50 MHz

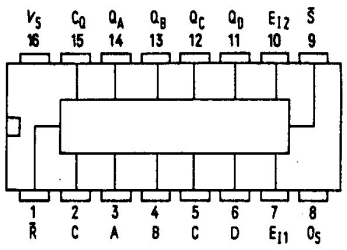
FLJ 381 - 74196 (Q67000 - J283)
FLJ 385 - 84196 (Q67000 - J292)

D. Counters and Frequency Dividers



Binary counter with set and reset for 50 MHz

FLJ 391 - 74197 (Q67000 - J284)
FLJ 395 - 84197 (Q67000 - J293)

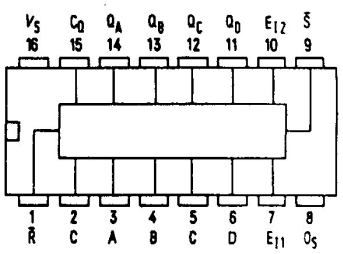


Synchronous decimal counter with synchronous set input and asynchronous reset input

FLJ 401 - 74160 (Q67000 - J277)
FLJ 405 - 84160 (Q67000 - J294)

Synchronous decimal counter with synchronous set and reset inputs

FLJ 421 - 74162 (Q67000 - J278)
FLJ 425 - 84162 (Q67000 - J296)

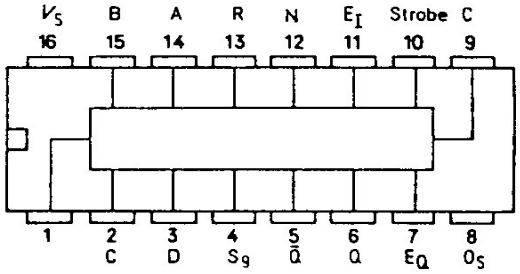


Synchronous 4-bit binary counter with synchronous set input and asynchronous reset input

FLJ 411 - 74161 (Q67000 - J273)
FLJ 415 - 84161 (Q67000 - J295)

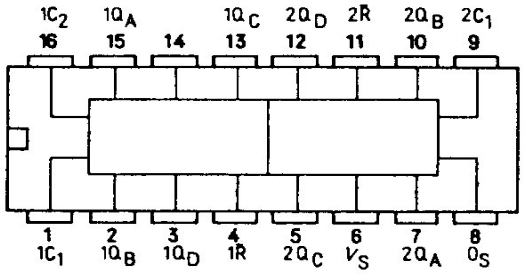
Synchronous 4-bit binary counter with synchronous set and reset inputs

FLJ 431 - 74163 (Q67000 - J279)
FLJ 435 - 84163 (Q67000 - J297)



Programmable decimal rate multiplier

FLJ 471 - 74167 (Q67000 - J308)

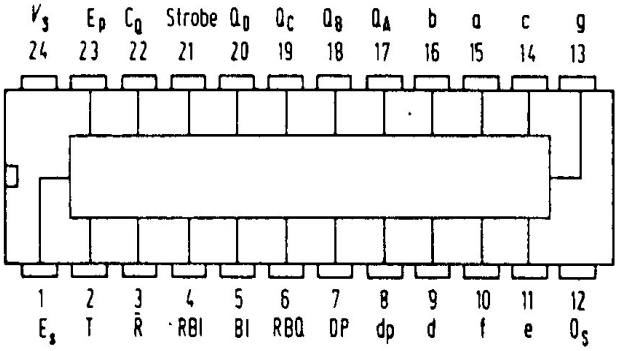


Dual binary counter for 50 MHz

FLJ 501 - 49704 (Q67000 - J371)
FLJ 505 - 49804 (Q67000 - J372)

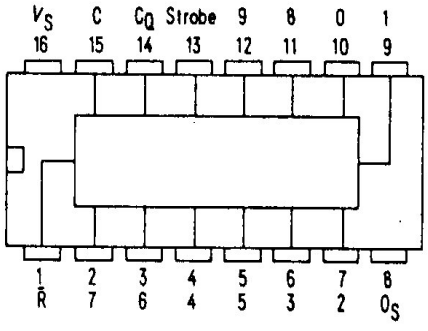
Dual decimal counter for 50 MHz

FLJ 511 - 49705 (Q67000 - J373)
FLJ 515 - 49805 (Q67000 - J374)



4-Bit binary counter, memory, 7-segment decoder and driver

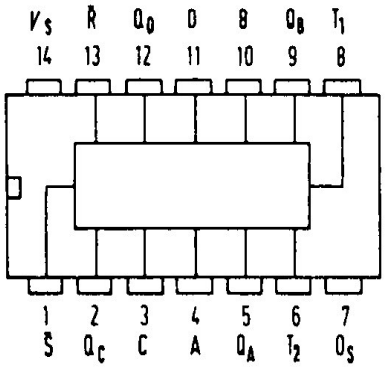
FLL 171 - 74143 (Q67000 - L58)
FLL 171 T - 74144 (Q67000 - L77)



Decimal counter, memory, decoder and driver for indicator tubes

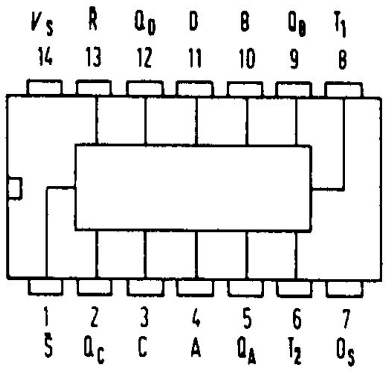
FLL 151 - 74142 (Q67000 - L57)

D. Counters and Frequency Dividers



Decimal counter for 35 MHz

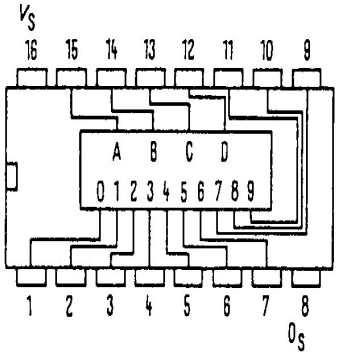
74176 (Q67000 - J434)
84176 (Q67000 - J480)



4-Bit binary counter for 35 MHz

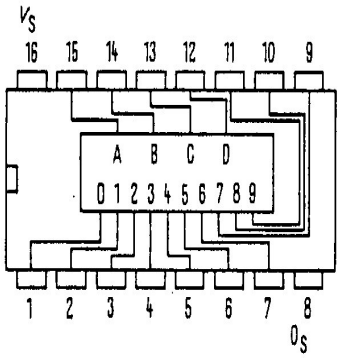
74177 (Q67000 - J435)
84177 (Q67000 - J481)

E. Decoders and Drivers



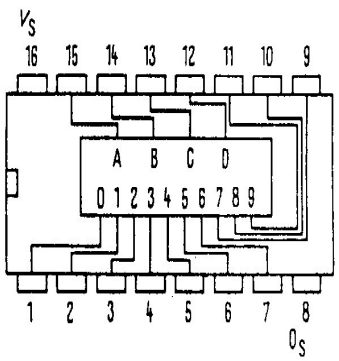
BCD decimal decoder

FLH 281 - 7442 (Q67000 - L3)
FLH 285 - 8442 (Q67000 - L13)



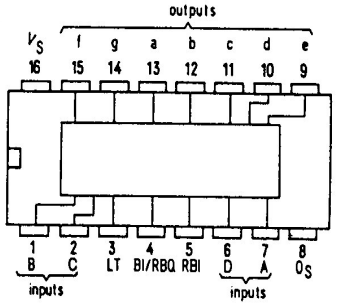
Excess-3 decimal decoder

FLH 361 - 7443 (Q67000 - H356)
FLH 365 - 8443 (Q67000 - H404)



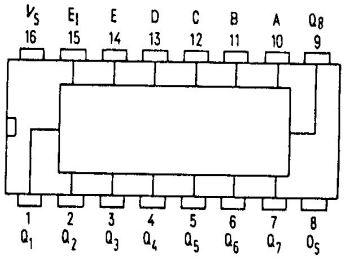
Excess-3-Gray decimal decoder

FLH 371 - 7444 (Q67000 - H357)
FLH 375 - 8444 (Q67000 - H403)



BCD-7-segment decoder

FLH 551 - 7448 (Q67000 - L12)
FLH 555 - 8448 (Q67000 - H448)

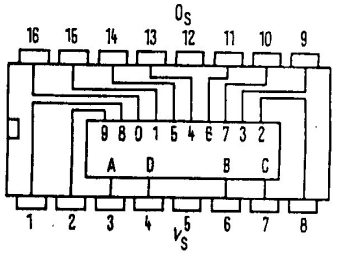


6-Bit-BCD binary converter

FLH 561 - 74184 (Q67000 - H619)
FLH 565 - 84184 (Q67000 - H592)

6-Bit-BCD binary converter

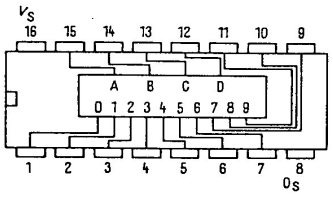
FLH 571 - 74185 A (Q67000 - H605)
FLH 575 - 84185 A (Q67000 - H622)



BCD decimal decoder-driver for indicator tubes

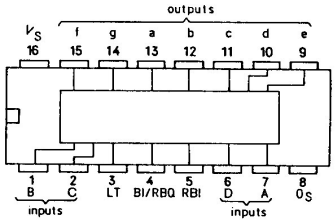
FLL 101 - 74141 (Q67000 - L1)

E. Decoders and Drivers



BCD decimal decoder-driver
with open collector output
for 80 mA and 30 V and 15 V, respectively

FLL 111 - 7445 (Q67000 - L9)
FLL 111 T - 74145 (Q67000 - L8)
FLL 115 - 8445 (Q67000 - L46)
FLL 115 T - 84145 (Q67000 - L45)



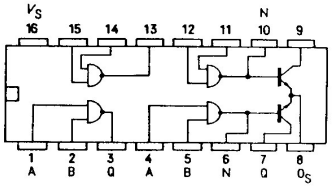
FLL 121; BCD 7-segment decoder
and driver with open collector output
with 30 V/20 mA

FLL 121 T; BCD 7-segment decoder and
driver with open collector output
with 15 V/20 mA

FLL 121 U; BCD 7-segment decoder and
driver with open collector output
with 30 V/40 mA

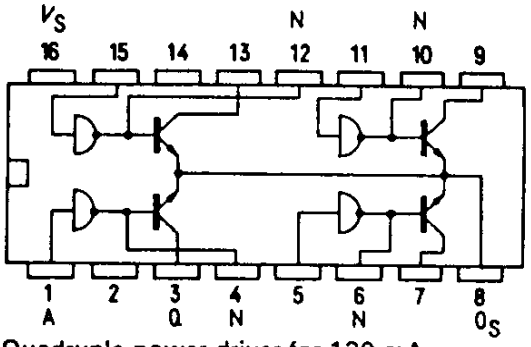
FLL 121 V; BCD 7-segment decoder and
driver with open collector output
with 15 V/40 mA

FLL 121 - 7446 (Q67000 - L10)
FLL 121 T - 7447 (Q67000 - L11)
FLL 121 U - 7446 A (Q67000 - L51)
FLL 121 V - 7447 A (Q67000 - L63)



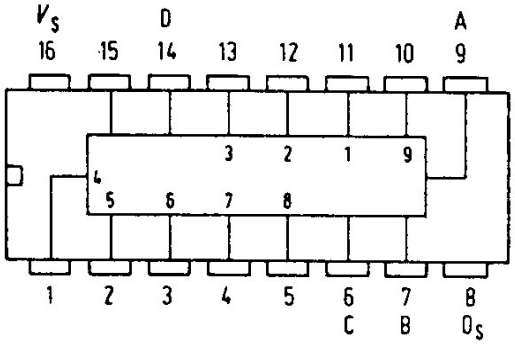
Dual AND power driver for 400 mA
and 30 V and 60 V, respectively,
and dual 2-input NAND gate

FLL 131 - 49700 (Q67000 - H236)
FLL 131 T - 49700 S1 (Q67000 - H668)
FLL 135 - 49800 (Q67000 - H314)
FLL 135 T - 49800 S1 (Q67000 - L67)



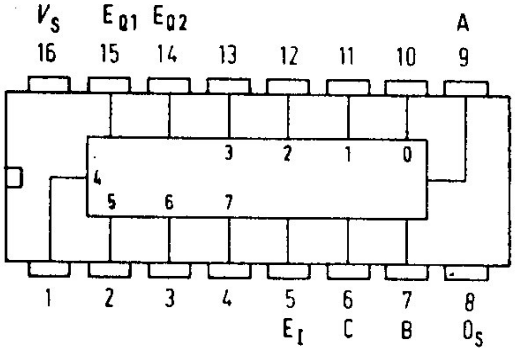
Quadruple power driver for 130 mA
and 30 V and 60 V, respectively

FLL 141 - 49701 (Q67000 - H331)
FLL 145 - 49801 (Q67000 - H415)
FLL 141 T - 49701 S1 (Q67000 - L92)



4-Bit decimal BCD converter

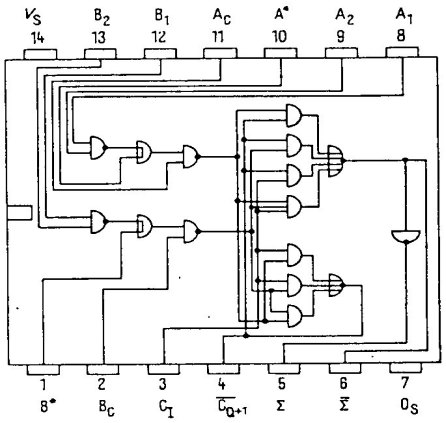
74147 (Q67000 - H724)
84147 (Q67000 - H854 - F9)



3-Bit decimal BCD converter

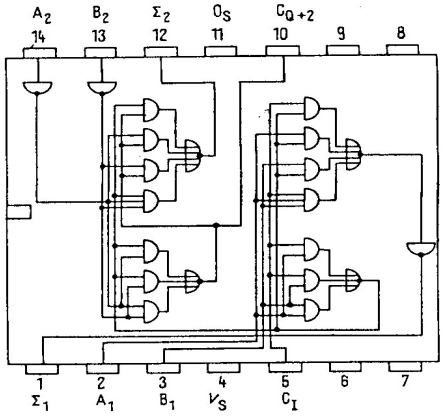
74148 (Q67000 - H725)
84148 (Q67000 - H855)

F. Adders



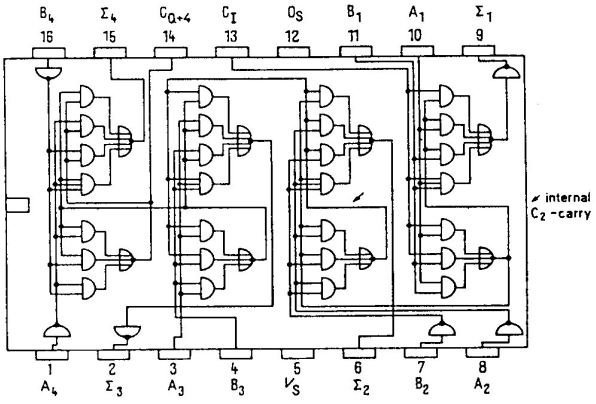
1-Bit full adder

FLH 221 - 7480 (Q67000 - H455)
FLH 225 - 8480 (Q67000 - H244)



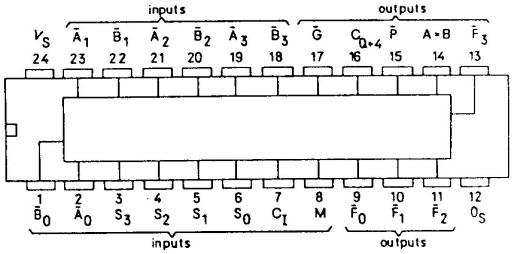
2-Bit full adder

FLH 231 - 7482 (Q67000 - H456)
FLH 235 - 8482 (Q67000 - H245)



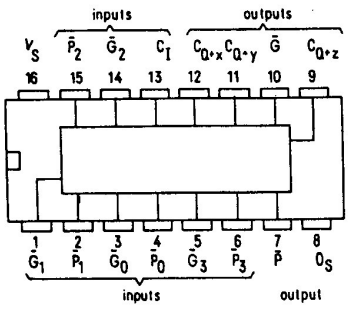
4-Bit full adder

FLH 241 - 7483 (Q67000 - H457)
FLH 245 - 8483 (Q67000 - H246)



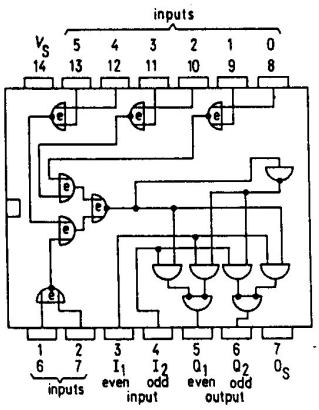
4-Bit arithmetic logic unit (ALU)

FLH 401 - 74181 (Q67000 - H431)
FLH 405 - 84181 (Q67000 - H427)



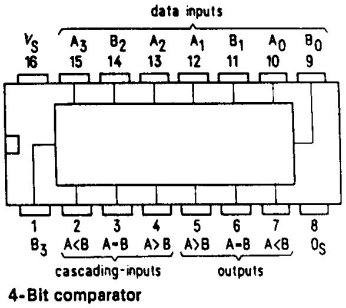
Look-ahead carry-generator for ALU

FLH 411 - 74182 (Q67000 - H486)
FLH 415 - 84182 (Q67000 - H501)



8-Bit parity generator

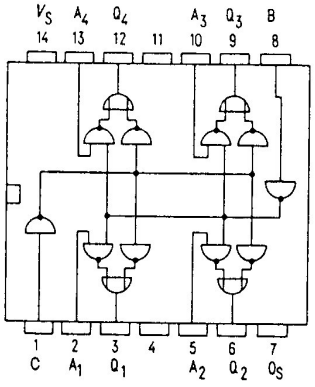
FLH 421 - 74180 (Q67000 - H299)
FLH 425 - 84180 (Q67000 - H453)



4-Bit comparator

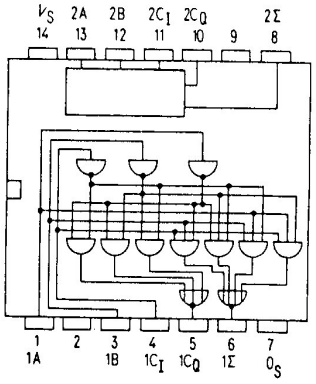
FLH 431 - 7485 (Q67000 - H494)
FLH 435 - 8485 (Q67000 - H508)

F. Adders



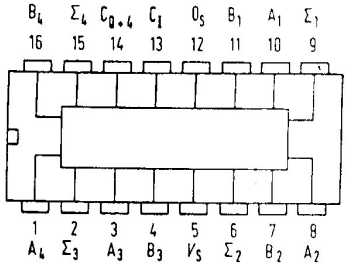
4-Bit complement unit

FLH 441 - 74 H87 (Q67000 - H324)
FLH 445 - 84 H87 (Q67000 - H510)



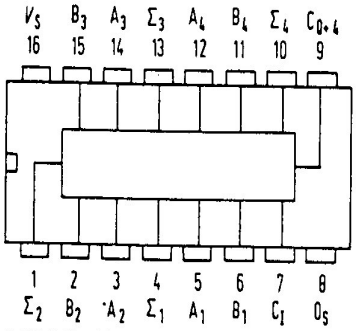
Dual 1-bit full adder

FLH 451 - 74 H183 (Q67000 - H495)
FLH 455 - 84 H183 (Q67000 - H511)



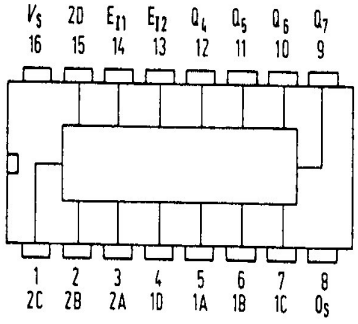
4-Bit full adder

7483 A (Q67000 - H849)
8483 A (Q67000 - H850)



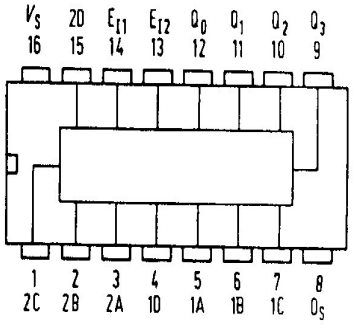
4-Bit full adder

74283 (Q67000 - H728)
84283 (Q67000 - H906)



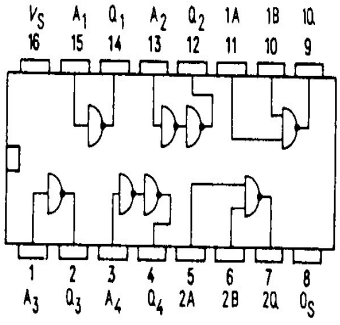
Dual 4-bit parallel multiplier

74284 (Q67000 - H924)
84284 (Q67000 - H917)



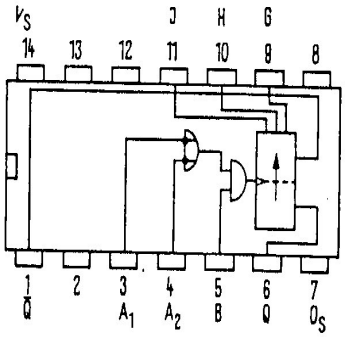
74285 (Q67000 - H926)
84285 (Q67000 - H918)

G. Miscel-laneous



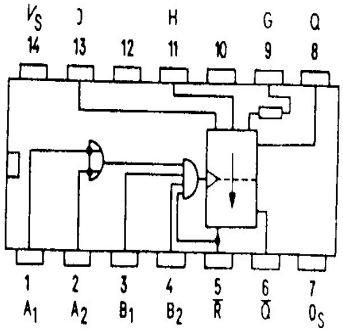
Delay unit

FLH 641 - 49703 (Q67000 - H604)



Monostable multivibrator

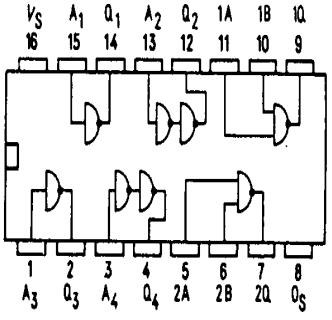
FLK 101 - 74121 (Q67000 - K13)
FLK 105 - 84121 (Q67000 - K10)



Retriggerable monostable multivibrator with reset

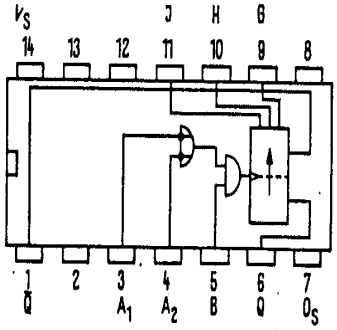
FLK 111 - 74122 (Q67000 - K27)
FLK 115 - 84122 (Q67000 - K28)

G. Miscel-laneous



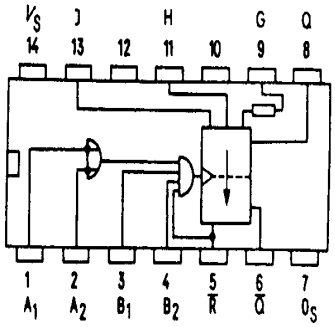
Delay unit

FLH 641 - 49703 (Q67000 - H604)



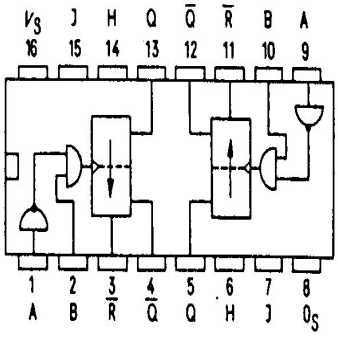
Monostable multivibrator

FLK 101 - 74121 (Q67000 - K13)
FLK 105 - 84121 (Q67000 - K10)



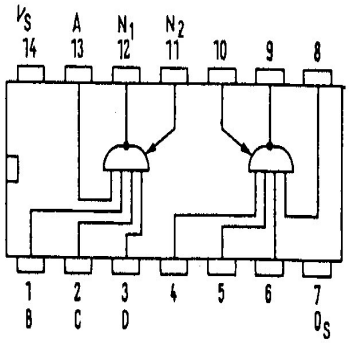
Retriggerable monostable multivibrator with reset

FLK 111 - 74122 (Q67000 - K27)
FLK 115 - 84122 (Q67000 - K28)



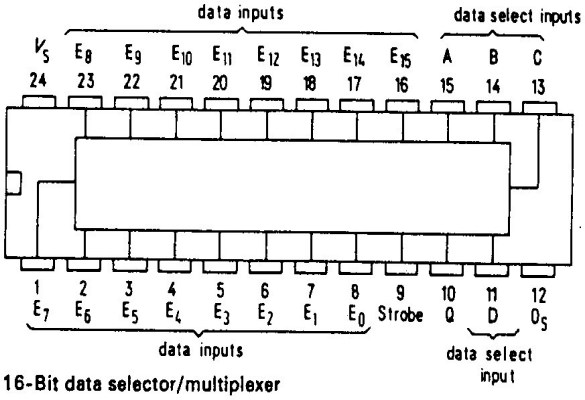
Dual retriggerable monostable multivibrator with reset

FLK 121 - 74123 (Q67000 - K29)
FLK 125 - 84123 (Q67000 - K30)



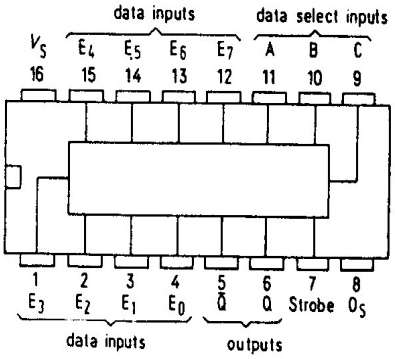
Dual 4-input expander

FLY 101 - 7460 (Q67000 - Y1)
FLY 105 - 8460 (Q67000 - Y18)



16-Bit data selector/multiplexer

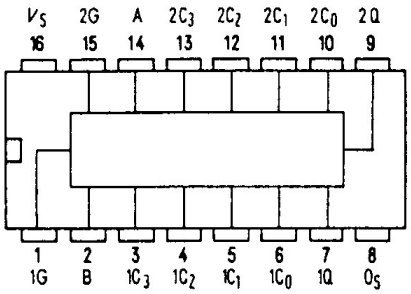
FLY 111 - 74150 (Q67000 - Y31)
FLY 115 - 84150 (Q67000 - Y46)



8-Bit data selector/multiplexer

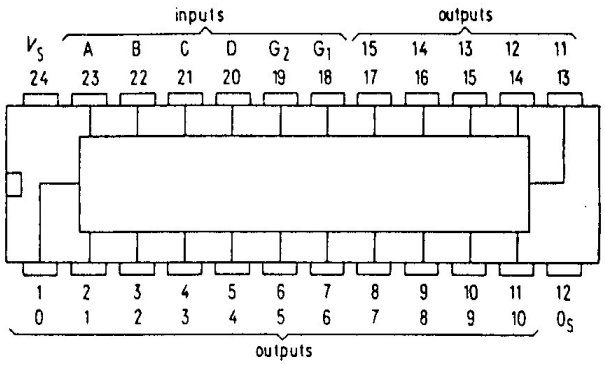
FLY 121 - 74151 (Q67000 - Y30)
FLY 125 - 84151 (Q67000 - Y29)

G. Miscel-laneous



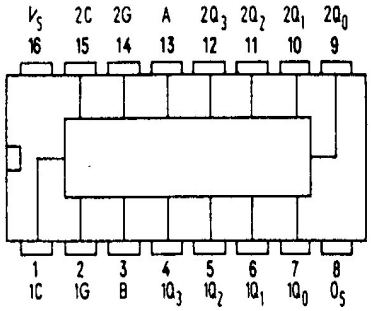
Dual 4-bit data selector/multiplexer

FLY 131 - 74153 (Q67000 - Y54)
FLY 135 - 84153 (Q67000 - Y58)



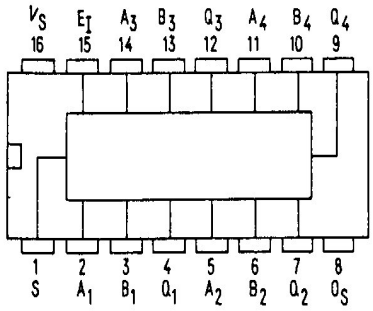
4-Bit binary decoder/demultiplexer

FLY 141 - 74154 (Q67000 - Y45)
FLY 145 - 84154 (Q67000 - Y51)



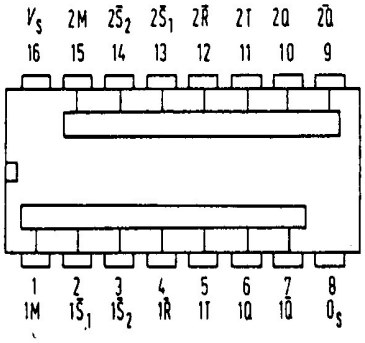
Dual 2-bit binary decoder/demultiplexer

FLY 151 - 74155 (Q67000 - Y52)
FLY 155 - 84155 (Q67000 - Y59)
FLY 161 - 74156 (Q67000 - Y55)
FLY 165 - 84156 (Q67000 - Y60)



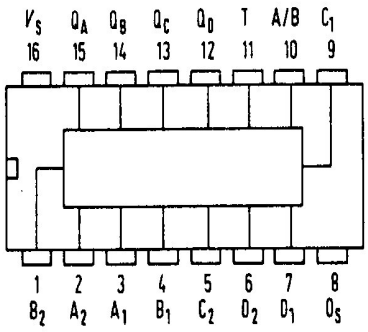
Quadruple 2-bit data selector/multiplexer

FLY 171 - 74157 (Q67000 - Y75)
FLY 175 - 84157 (Q67000 - Y77)



Dual pulse synchronizer

FLY 181 - 74120 (Q67000 - Y98)
FLY 185 - 84120 (Q67000 - Y127)



Quadruple 2-bit data selector with memory

74298 (Q67000 - Y162)
84298 (Q67000 - Y163)

11.6 LSL-Series (Low-Speed Noise-Immune Logic)

General information on the FZ 100-LSL Serie

FZ 100 is a low-speed noise-immune logic series of monolithic IC's. Their use in applications where proper functioning may be impaired by noise pulses offers the following advantages:

1. Higt static noise immunity.
   Depending upon switching condition, its tupical range lies between 5 and 8 V using a Z-diode at an operating voltage 15 V.

2. Higt dynamic noise immunity.
   It is achieved by a high pulse duration and long propagation delay times (typ. 350 ns).

3. Variable extension of switching times by using additional capacitors between the N and Q pins. The dynamic noise immunity may thus be increased, as required.

4. Operating voltage range between 11.4 and 17 V.

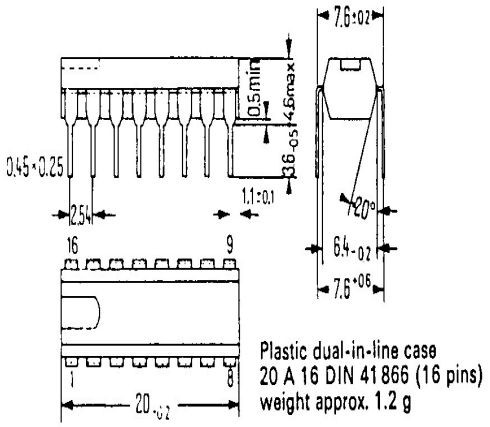
5. push-pull output stage with a low output resistance in both states.

6. Plastic dual-in-line package.

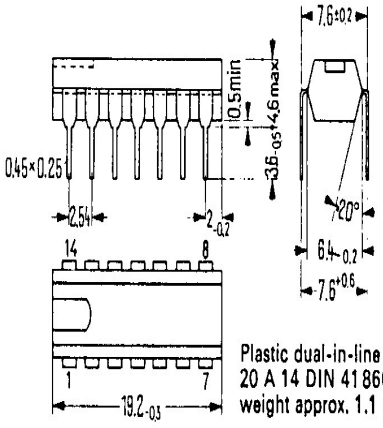
7. Two temperature ranges: 0 to 70 °C (range 1, e. g. FZH 101); -25 to 85 °C (range 5, e. g. FZH 105).

The following maximum ratings apply to all types:		Min	Max	Unit
Supply voltage	V_S	0	18	V
Input voltage	V_{in}	0	18	V
Operating temperature (range 1)	T_{amb}	0	70	°C
Operating temperature (range 5)	T_{amb}	-25	85	°C
Storage temperature	T_S	-65	150	°C
Thermal ressitance (system to ambient air)	R_{thSamb}		150	K/W

LSL series' package outlines and dimensional drawings:

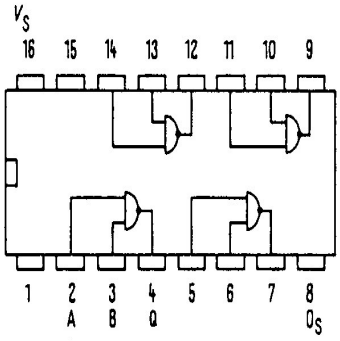


Plastic dual-in-line case
20 A 16 DIN 41866 (16 pins)
weight approx. 1.2 g



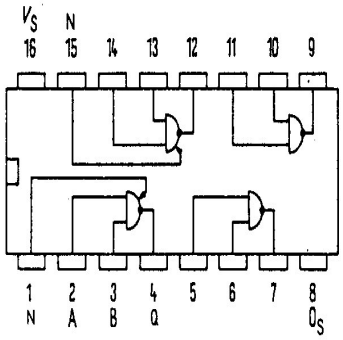
Plastic dual-in-line case
20 A 14 DIN 41866 (14 pins)
weight approx. 1.1 g, dimensions in mm

A. Gates



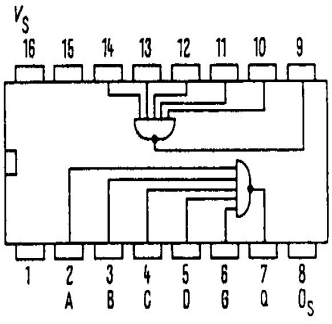
Quadruple 2-input NAND gate

FZH 101 (Q67000 - H190)
FZH 105 (Q67000 - H250)



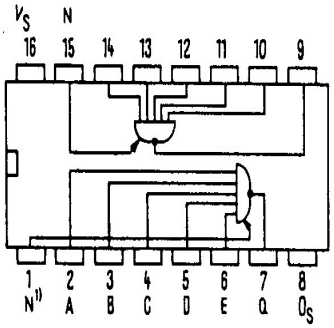
Quadruple 2-input NAND gate with N-input

FZH 111 (Q67000 - H191)
FZH 115 (Q67000 - H215)



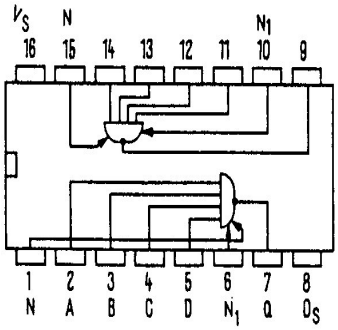
Dual 5-input NAND gate

FZH 121 (Q67000 - H192)
FZH 125 (Q67000 - H254)



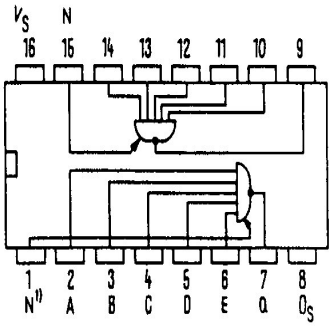
Dual 5-input NAND gate with N-input

FZH 131 (Q67000 - H193)
FZH 135 (Q67000 - H255)



Dual 4-input NAND gate with expander node N_1 and N-input

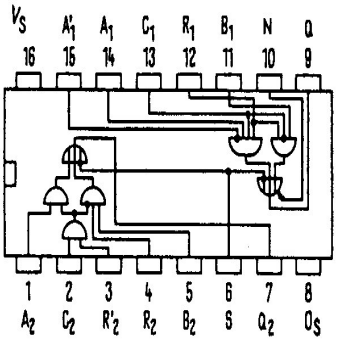
FZH 171 (Q67000 - H328)
FZH 175 (Q67000 - H329)



Dual 5-input NAND power gate with N-input

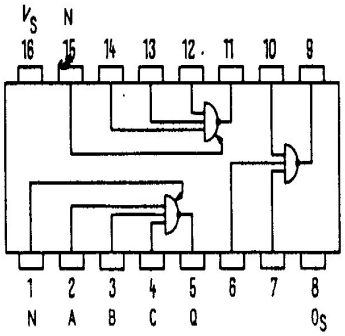
FZH 141 (Q67000 - H194)
FZH 145 (Q67000 - H256)

A. Gates



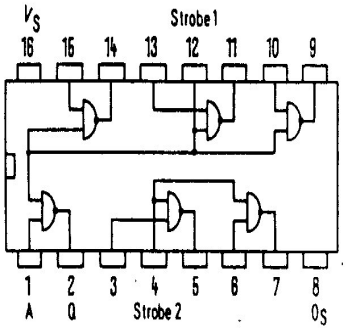
Dual AND/OR gate
with N-input

FZH 151 (Q67000 - H195)
FZH 155 (Q67000 - H260)



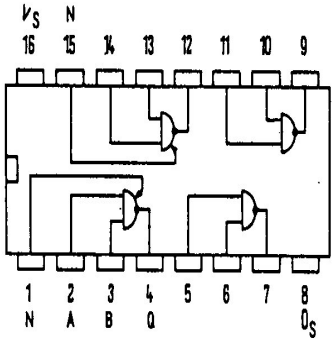
Triple 3-input NAND gate
with N-input

FZH 191 (Q67000 - H633)
FZH 195 (Q67000 - H634)



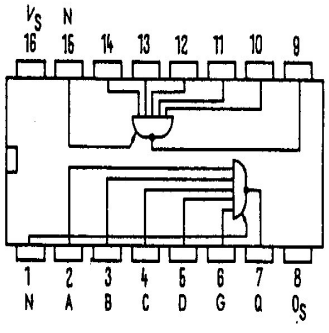
Hex inverter with strobe input

FZH 201 (Q67000 - H636)
FZH 205 (Q67000 - H637)



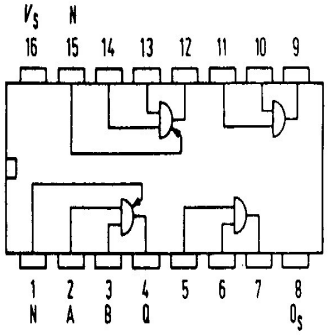
Quadruple 2-input NAND gate with
open collector output and N-input

FZH 211 (Q67000 - H639)
FZH 215 (Q67000 - H640)



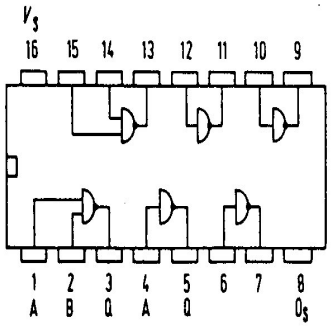
Dual 5-input NAND gate with open
collector output and N-input

FZH 231 (Q67000 - H642)
FZH 235 (Q67000 - H643)



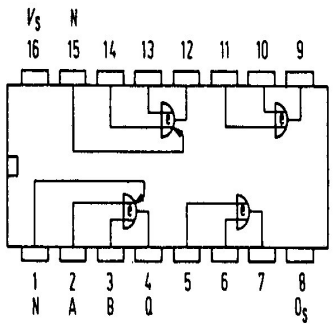
Quadruple 2-input AND gate
with N-input

FZH 251 (Q67000 - H817)
FZH 255 (Q67000 - H818)



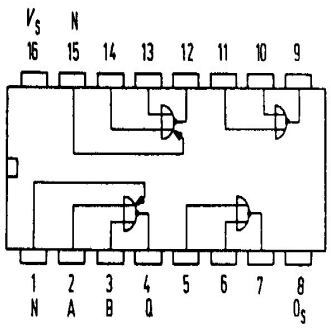
Dual 2-input NAND gate
and quadruple inverter

FZH 261 (Q67000 - H819)
FZH 265 (Q67000 - H820)



Quadruple 2-input exclusive-OR gate
with N-input

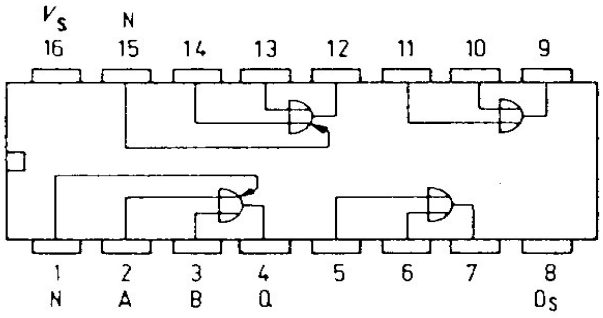
FZH 271 (Q67000 - H821)
FZH 275 (Q67000 - H822)



Quadruple 2-input NOR gate
with N-input

FZH 281 (Q67000 - H823)
FZH 285 (Q67000 - H824)

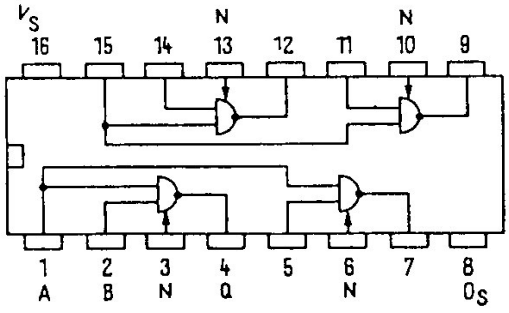
A. Gates



Quadruple 2-input OR gate with N-input

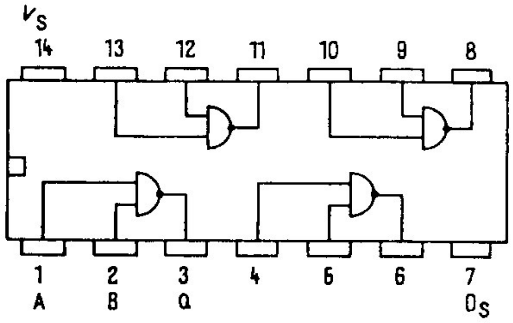
FZH 291 (Q67000 - H825)
FZH 295 (Q67000 - H826)

B. Level Converters



Quadruple LSL-TTL level converter with open collector output and N-input

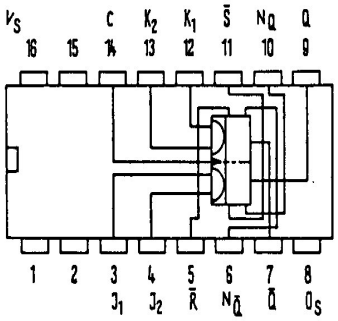
FZH 161 (Q67000 - H288)
FZH 165 (Q67000 - H289)



Quadruple TTL-LSL level converter with open collector output

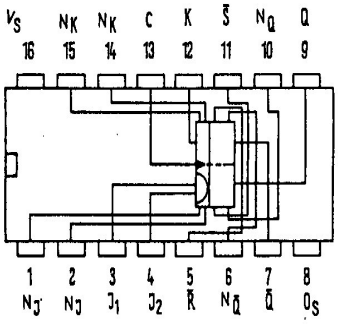
FZH 181 (Q67000 - H326)
FZH 185 (Q67000 - H327)

C. Flipflops and Adders



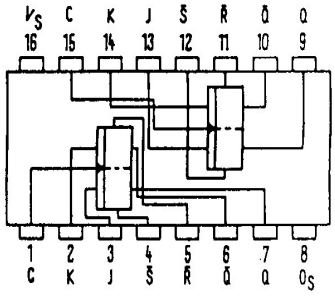
JK-master-slave-flipflop with 2 J and K-inputs and N-inputs

FZJ 101 (Q67000 - J95)
FZJ 105 (Q67000 - J124)



JK-master-slave-flipflop with N-inputs

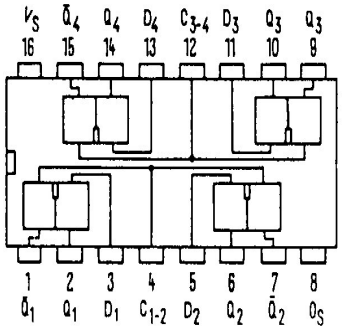
FZJ 111 (Q67000 - J96)
FZJ 115 (Q67000 - J125)



Dual JK-master-slave-flipflop with set and reset

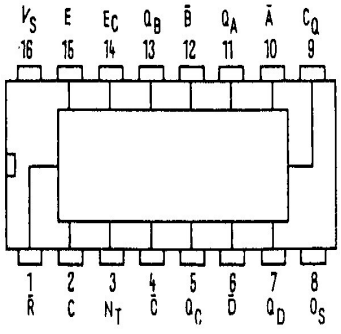
FZJ 121 (Q67000 - J385)
FZJ 125 (Q67000 - J386)

C. Flipflops and Adders



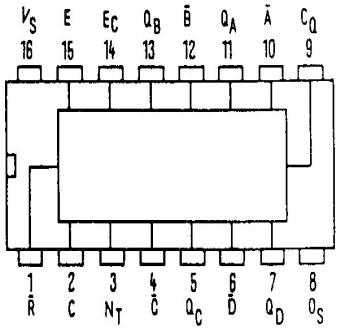
Quadruple D-flipflop

FZJ 131 (Q67000 - J388)
FZJ 135 (Q67000 - J389)



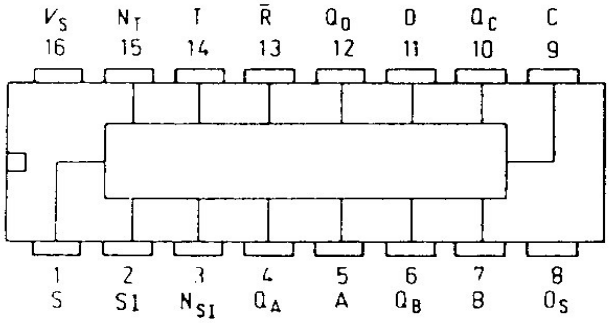
Synchronous decimal counter with set and reset, clock and carry gating

FZJ 141 (Q67000 - J391)
FZJ 145 (Q67000 - J392)



Synchronous binary counter with set and reset, clock and carry gating

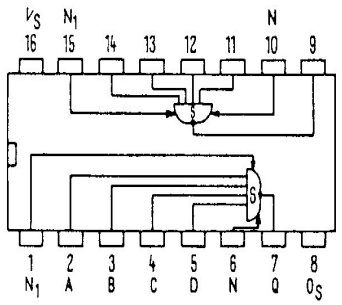
FZJ 151 (Q67000 - J394)
FZJ 155 (Q67000 - J395)



Synchronous 4-bit shiftregister with set and reset and N-input

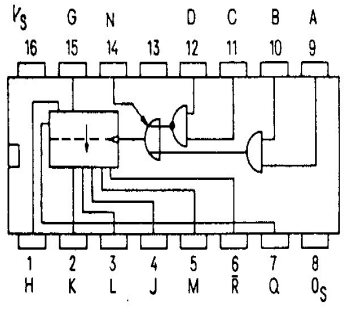
FZJ 161 (Q67000 - J507)
FZJ 165 (Q67000 - J562)

D. Special Circuits



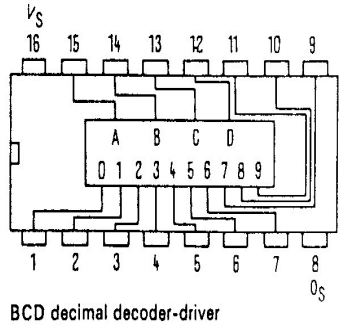
Dual 4-input NAND Schmitt trigger with expander node N_1 and N-input

FZH 241 (Q67000 - H645)
FZH 245 (Q67000 - H646)



Timing circuit for the following operating modes: monostable multivibrator, pulse delay, pulse reduction

FZK 101 (Q67000 - K6)
FZK 105 (Q67000 - K7)



BCD decimal decoder-driver for indicator tubes

FZL 101 (Q67000 - L68)

REFERÊNCIAS BIBLIOGRÁFICAS

CAPUANO, Francisco Gabriel; IDOETA, Ivan Valeije. **Elementos de Eletrônica Digital.** 28. Ed. São Paulo, Érica, 1998, 526p.

MALVINO, Albert Paul, LEACH, Donald P. **Eletrônica Digital Princípios e Aplicações;** lógica seqüencial. São Paulo, McGraw-Hill, 1987, 2 v.

MOTOROLA **CMOS Logic Data**. USA, 1990.

_____ . **Fast and LS TTL.** USA, 1989.

RCA Solid State. **COS/MOS**; Digital Integrated Circuits. USA, 1978.

SIEMENS. **Discrete Semeconductors, Integrated Circuits, Power Semiconductors**; delivery program 1973/74. USA.

SENAI. CE. CFP WDS. **Prática de Eletrônica Digital,** lógica combinacional e lógica seqüencial. Fortaleza, 1997, 40 p.

EQUIPE TÉCNICA

COORDENAÇÃO: Leônidas Fernandes Macedo Júnior

ELABORADORES: Francisco Álvaro da Silva Júnior
Marcos Paulo Souza Silva

DIGITAÇÃO E DIAGRAMAÇÃO: Sandra Lúcia Carvalho de Araújo

CORREÇÃO ORTOGRÁFICA: Rosângela Fátima Carvalho de Araújo

NORMALIZAÇÃO BIBLIOGRÁFICA: Núcleo de Informação Tecnológica - NIT